

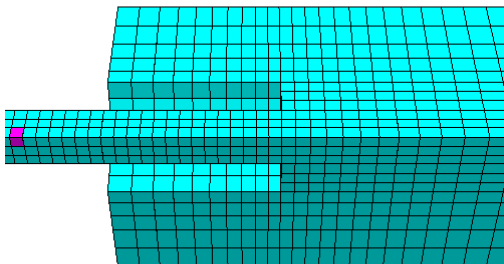
## JRC TECHNICAL REPORTS

# Shock tube tests with coupled 1D-3D models in EUROPLEXUS

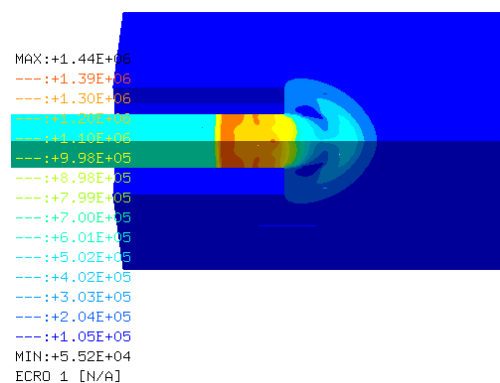
F. Casadei  
V. Aune  
F. Daude  
P. Galon  
G. Valsamos  
M. Larcher

2016

SHOC55  
TIME: 0.00000E+00 STEP: 0



SHOC55  
TIME: 2.30017E-02 STEP: 1410





Shock tube tests with coupled 1D-3D  
models in EUROPLEXUS

This publication is a Technical report by the Joint Research Centre, the European Commission's in-house science service. It aims to provide evidence-based scientific support to the European policy-making process. The scientific output expressed does not imply a policy position of the European Commission. Neither the European Commission nor any person acting on behalf of the Commission is responsible for the use which might be made of this publication.

**JRC Science Hub**

<https://ec.europa.eu/jrc>

JRC101011

EUR 27890 EN

ISBN 978-92-79-57976-9 (print)

ISSN 1831-9424 (print)

doi:10.2788/702507 (print)

© European Union, 2016

Reproduction is authorised provided the source is acknowledged.

Printed in Italy

All images © European Union 2016

How to cite: Authors; title; EUR; doi

# Shock tube tests with coupled 1D-3D models in EUROPLEXUS

F. Casadei<sup>1</sup>, V. Aune<sup>2</sup>, F. Daude<sup>3</sup>, P. Galon<sup>4</sup>, G. Valsamos<sup>5</sup> and M. Larcher<sup>5</sup>

<sup>1</sup>*Retired from JRC ELSA*

<sup>2</sup>*Structural Impact Laboratory (SIMLab)  
and Department of Structural Engineering, Norwegian University of Science and Technology,  
Rich. Birkelands vei 1A, NO-7491 Trondheim, Norway*

<sup>3</sup>*EDF R&D, AMA/T61*

<sup>4</sup>*CEA Saclay, DEN/DANS/DM2S/SEMT/DYN*

<sup>5</sup>*European Laboratory for Structural Assessment (ELSA),  
Institute for the Protection and Security of the Citizen (IPSC),  
Joint Research Centre (JRC), 21027 Ispra, Italy*

February 13, 2016

## Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>The test problem</b>	<b>2</b>
<b>3</b>	<b>Numerical tests</b>	<b>3</b>
3.1	Multi-dimensional solutions . . . . .	4
3.1.1	1D3D01 . . . . .	4
3.1.2	1D3D02 . . . . .	5
3.2	One-dimensional solutions . . . . .	6
3.2.1	1D3D03 . . . . .	6
3.3	Mixed-dimensional solutions . . . . .	6
3.3.1	1D3D14 . . . . .	7
3.3.2	1D3D15 . . . . .	9
3.3.3	1D3D16 . . . . .	10
3.3.4	1D3D17 . . . . .	11
3.4	Application to NTNU shock tube . . . . .	11
3.4.1	SHOC51 . . . . .	12
3.4.2	SHOC52 . . . . .	14
3.4.3	SHOC53 – Triggering the mesh adaptation . . . . .	15
3.4.4	SHOC54 . . . . .	15
3.4.5	SHOC55 . . . . .	17
3.4.6	SHOC61 . . . . .	18
3.4.7	SHOC62 . . . . .	21
3.4.8	SHOC65 . . . . .	21
<b>4</b>	<b>Conclusions and future work</b>	<b>22</b>
	<b>References</b>	<b>23</b>
	<b>Appendix A — Input files</b>	<b>24</b>
	<b>List of input files</b>	<b>42</b>

## List of Figures

1	The shock tube test rig. . . . .	3
2	The shock tube problem. . . . .	3
3	Some possible choices for mixed 1D-3D models. . . . .	4
4	Results of test 1D3D01. . . . .	5
5	Results of test 1D3D02. . . . .	5
6	Results of test 1D3D03. . . . .	7
7	Results of test 1D3D14. . . . .	10
8	Results of test 1D3D16. . . . .	11
9	Results of test 1D3D17. . . . .	11
10	Basic model of the NTNU test rig. . . . .	12
11	Some results of test SHOC51. . . . .	13
12	Further results of test SHOC51. . . . .	13
13	Some results of test SHOC52. . . . .	14
14	Further results of test SHOC52. . . . .	15
15	Some results of test SHOC53. . . . .	16
16	Further results of test SHOC53. . . . .	16
17	Some results of test SHOC55. . . . .	17
18	Further results of test SHOC55. . . . .	18
19	Mixed 1D-3D mesh of the test rig for case SHOC61. . . . .	18
20	Some results of test SHOC61. . . . .	20
21	Further results of test SHOC51. . . . .	20
22	Comparison of final pressures in cases SHOC52 and SHOC62 . . . . .	21
23	Further results of test SHOC62. . . . .	22
24	Comparison of final pressures in cases SHOC62 and SHOC65 . . . . .	22
25	Further results of test SHOC65. . . . .	23

## List of Tables

1	Multi-dimensional simulations. . . . .	4
2	One-dimensional simulations. . . . .	6
3	Mixed-dimensional simulations. . . . .	6
4	Shock tube simulations. . . . .	12

## 1 Introduction

This document presents some numerical tests for the verification of the coupling between 1D and 3D Cell Centred Finite Volume (VFCC) fluid models in the EUROPLEXUS code (EPX).

EUROPLEXUS [1] is a computer code jointly developed by the French Commissariat à l’Energie Atomique (CEA DMT Saclay) and by EC-JRC. The code application domain is the numerical simulation of fast transient phenomena such as explosions, crashes and impacts in complex three-dimensional fluid-structure systems.

The Cast3m [2] software from CEA is used as a pre-processor to EPX when it is necessary to generate complex meshes.

## 2 The test problem

The scope here is to verify the possibility of joining a 1D mesh with a 3D (or 2D) fluid mesh. In some experiments being conducted at NTNU Trondheim, a long shock tube is used to load some structures located near one extremity of the tube.

Figure 1 shows the NTNU test rig.

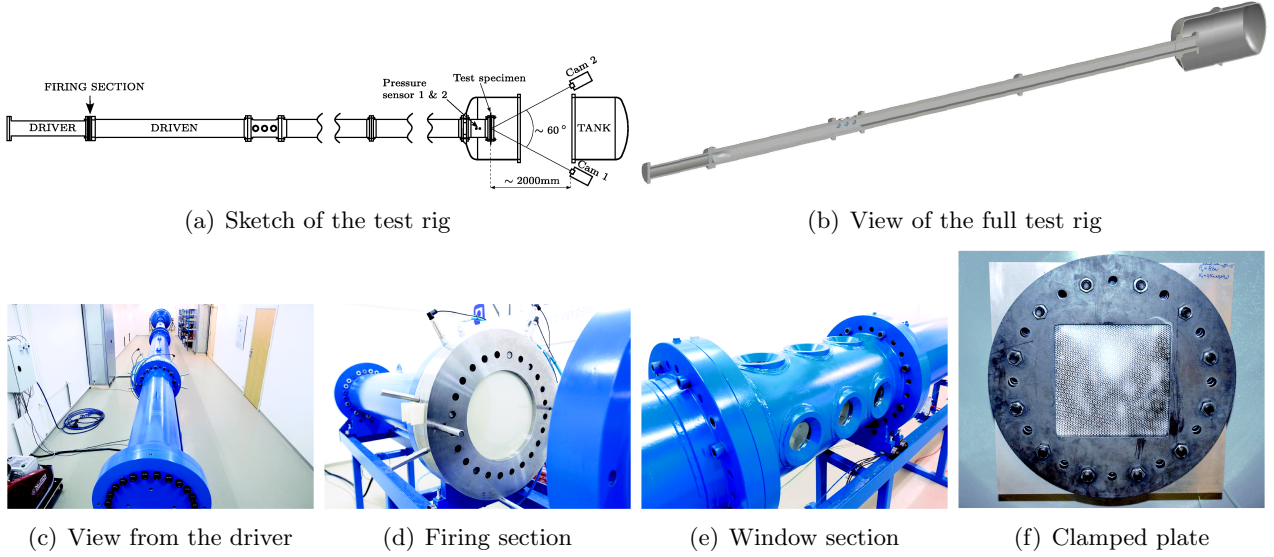


Figure 1: The shock tube test rig.

In the test zone (near the clamped plate), the problem is clearly multi-dimensional and a full 3D model is needed. However, the vast majority of the tube has a constant cross-section and lends itself ideally to being modelled in 1D, thus saving a lot of CPU time in the numerical simulations compared with a full 3D model of the whole test rig.

### 3 Numerical tests

The test problem considered here is a classical shock tube test, see Figure 2. A long tube with rigid walls is filled with a perfect gas, initially at rest. The tube is subdivided in two regions, which have different initial conditions in terms of initial density, and thus of initial pressure of the gas. At the initial time, the imaginary wall separating the two regions is instantly removed and a shock wave starts to propagate.

HP	LP
----	----

Figure 2: The shock tube problem.

This problem admits a well-known analytical solution, which is used as a reference for the numerical solutions obtained next.

The assumed initial conditions are as follows:

- **High-pressure** zone. Density  $\rho = 14.1 \text{ kg/m}^3$ , pressure  $p = 12 \text{ bar}$ , ratio of specific heats  $\gamma = C_p/C_v = 1.4$ , specific heat at constant volume  $C_v = 720 \text{ J/kg}\cdot\text{K}$ , reference pressure  $P_{\text{ref}} = 1 \text{ bar}$ .
- **Low-pressure** zone. Density  $\rho = 1.2 \text{ kg/m}^3$ , pressure  $p = 1 \text{ bar}$ , ratio of specific heats  $\gamma = C_p/C_v = 1.4$ , specific heat at constant volume  $C_v = 720 \text{ J/kg}\cdot\text{K}$ , reference pressure  $P_{\text{ref}} = 1 \text{ bar}$ .

Since the problem is self-similar, the length of the two zones has no influence on the solution (nor has an influence the cross-section of the tube, provided it is constant). We assume here a length of  $L_{HP} = L_{LP} = 16 \text{ m}$  for both zones, simply in order to avoid reflected waves at the extremities for the entire duration of the test of interest,  $t_f = 20 \text{ ms}$ . The assumed mesh size is  $h = 10 \text{ mm}$ , so there are 3200 elements along the tube, altogether.

In the following, we first obtain a purely 3D solution of the test problem, and compare it with the reference analytical solution. Then, a purely 1D solution is obtained and compared. Finally, an attempt is made to obtain a mixed 1D-3D solution in which one (or more) parts of the tube are meshed in 1D and the rest is modelled in 3D, see e.g. Figure 3.

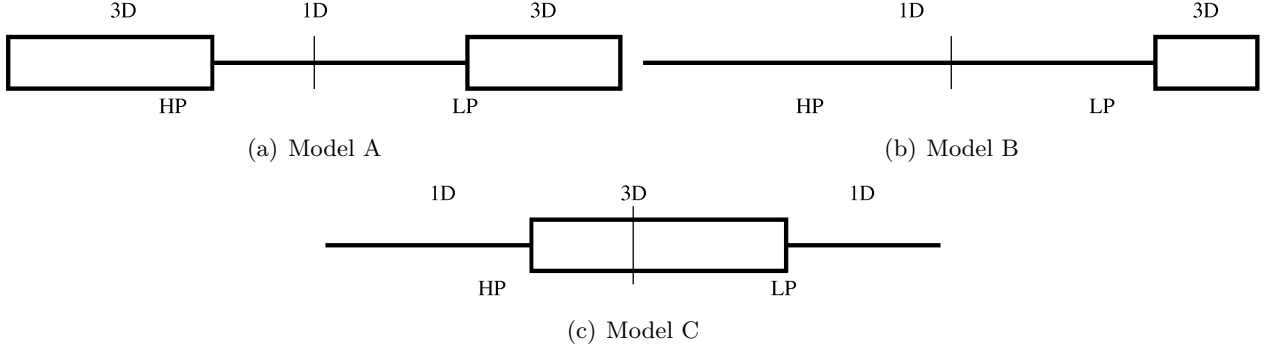


Figure 3: Some possible choices for mixed 1D-3D models.

### 3.1 Multi-dimensional solutions

Purely multi-dimensional solutions are obtained either in 3D or in 2D. The calculations done are summarized in Table 1.

Case	Description	CPU [s]
1D3D01	3D mesh, CUVF cube elements	34.3
1D3D02	2D mesh, Q4VF quadrilateral elements	18.3

Table 1: Multi-dimensional simulations.

#### 3.1.1 1D3D01

The EPX input file for this first calculation is listed and shortly commented below.

```

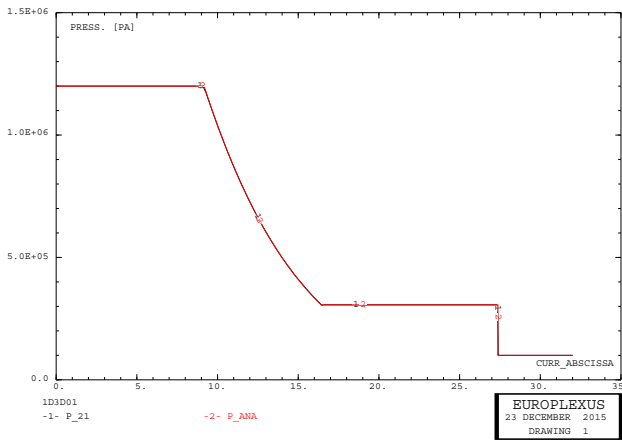
1D3D01
ECHO
!CONV win
CAST mesh
TRID EULE
GEOM CUVF flui TERM
COMP COUL ROUG LECT hp TERM
TURQ LECT lp TERM
MATE GAZP RO 14.4 GAMM 1.4 CV 720 PINI 12.E5 PREF 1.E5
      LECT hp TERM
      GAZP RO 1.20 GAMM 1.4 CV 720 PINI 1.0E5 PREF 1.E5
      LECT lp TERM
ECRI TFRE 1.E-3
      NOPO NOEL
      FICH ALIC TFRE 1.E-3
OPTI NOTE CSTA 0.75
      LOG 1
      VFCC FCON 6 ! hllc solver
      ORDR 2 ! order in space
      UTPS 2 ! order in time
      RECO 1
CALC TINI 0 TEND 20.E-3
FIN

```

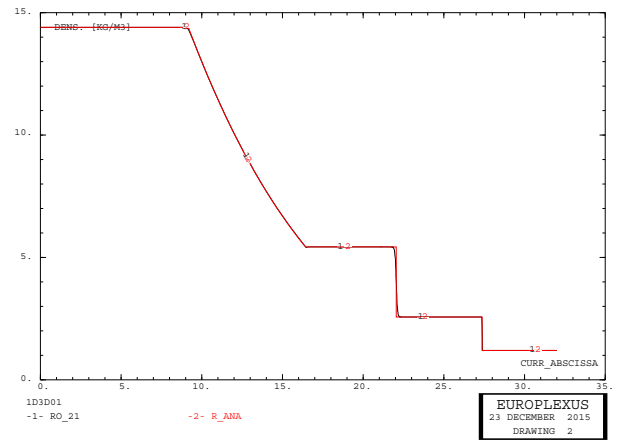
Since the tube walls are rigid, there is no need to represent the structure. Only the fluid is meshed. The chosen formulation is Eulerian. For the VFCC, the HLLC flux solver (which is the default one) is chosen. A second-order in space (ORDR 2 RECO 1) and in time (UTPS 2) solution is activated, to obtain the best possible accuracy.

The results of this simulation in terms of pressure, density and velocity distributions along the tube at the final time are compared in Figure 4 against the analytical solution (red curves), showing outstanding agreement.

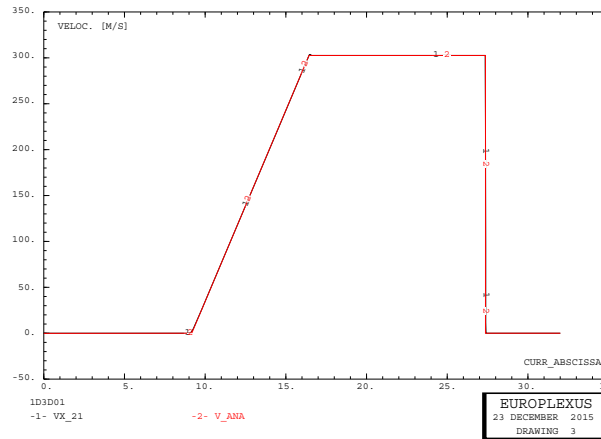




(a) Pressure



(b) Density



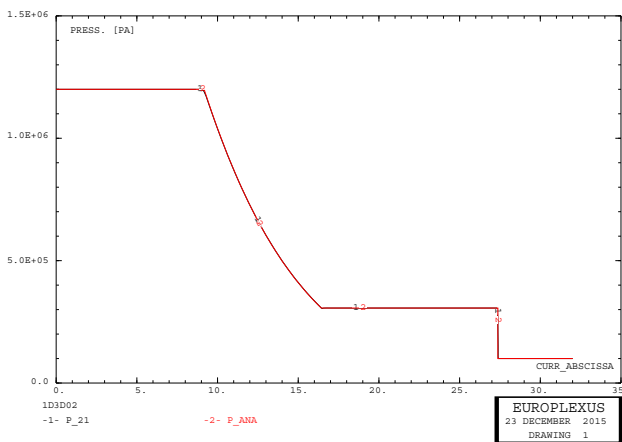
(c) Velocity

Figure 4: Results of test 1D3D01.

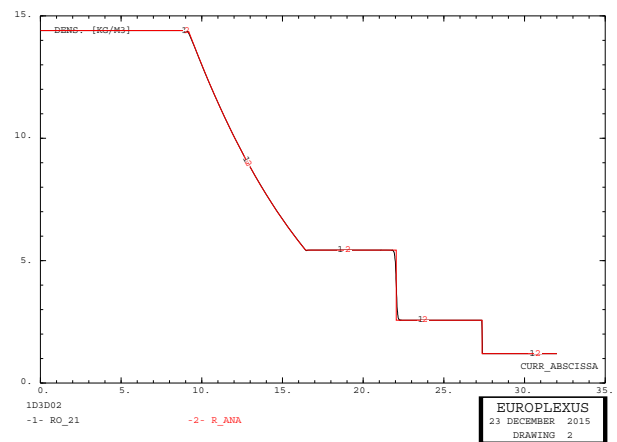
### 3.1.2 1D3D02

This test is similar to 1D3D01 but uses the Q4VF quadrilateral finite volume element in 2D.

The solution obtained is practically identical to the previous one, and in excellent agreement with the reference, see Figure 5.



(a) Pressure



(b) Density

Figure 5: Results of test 1D3D02.

### 3.2 One-dimensional solutions

Now we try obtaining a purely one-dimensional numerical solution. The calculations done are summarized in Table 2.

Case	Description	CPU [s]
1D3D03	1D mesh, TUVF segment elements	6.2

Table 2: One-dimensional simulations.

#### 3.2.1 1D3D03

A first attempt at using the TUVF element with exactly the same options as in the previous simulations failed because the code gives an error message stating that RECO 1 is not available for 1D VFCCs. This means that full second-order solution in space cannot be achieved with the TUVF mesh.

The EPX input file for this first calculation is listed and shortly commented below. Note that the RECO 1 option has been commented out.

```
1D3D03
ECHO
!CONV win
CAST mesh
TRID EULE
GEOM TUVF flui TERM
COMP DIAM DROI 0.01 LECT flui TERM
      COUL ROUG LECT hp TERM
      TURQ LECT lp TERM
MATE GAZP RO 14.4 GAMM 1.4 CV 720 PINI 12.E5 PREF 1.E5
      LECT hp TERM
      GAZP RO 1.20 GAMM 1.4 CV 720 PINI 1.0E5 PREF 1.E5
      LECT lp TERM
ECRI TFRE 1.E-3
      NOPO NOEL
      FICH ALIC TFRE 1.E-3
OPTI NOTE CSTA 0.75
      LOG 1
      VFCC FC0N 6 ! hllc solver
      ORDR 2 ! order in space
      OTPS 2 ! order in time
!      RECO 1 ! Not accepted by CAL_VFCC_1D
CALC TINI 0 TEND 20.E-3
FIN
```

The solution obtained is shown in Figure 6. Although the solution is in very good agreement with the reference, the quality is slightly less than in the previous solutions due to the lack of second order in space accuracy.

### 3.3 Mixed-dimensional solutions

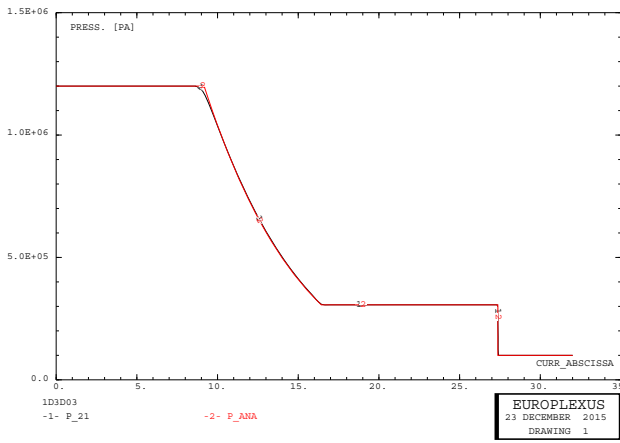
We now try to obtain solutions using a mixed model, partly meshed by 1D finite volumes TUVF and partly by 3D finite volumes CUVF. Although the final desired mixed model in the real application will be similar to the one shown in Figure 3(b) (with one zone in 1D and one zone in 3D), we consider rather a model similar to Figure 3(a) which has two 3D zones and a single 1D zone. This will make the comparison with the analytical and with the fully 1D solutions simpler.

In this model, the left (high-pressure) 3D zone has a length of 12 m while the right (low-pressure) 3D zone has a length of 10 m. The central zone (partly in the high-pressure and partly in the low-pressure region) meshed in 1D has a length of 10 m. Only one element across the tube diameter is used in the 3D zones.

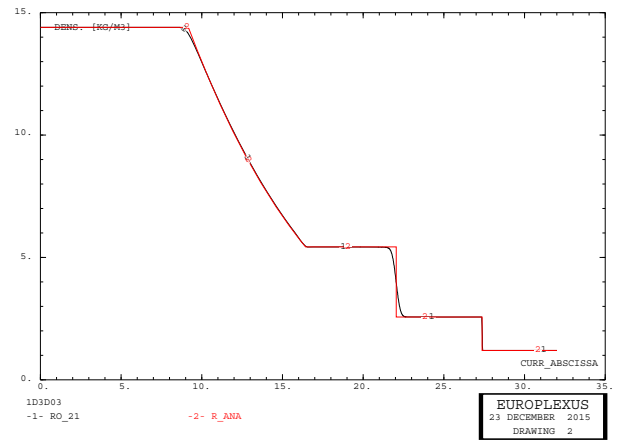
The calculations done are summarized in Table 3.

Case	Description	CPU [s]
1D3D14	1D-3D mesh, TUVF segment and CUVF cube elements	15.2
1D3D15	Same as 1D3D14 but alternative choice of DIAM, DTUB, COEF	15.7
1D3D16	Same as 1D3D14 but TUBE (FE) instead of TUVF (CCFV)	16.8
1D3D17	Same as 1D3D16 but RECO 1 (second order in space)	25.9

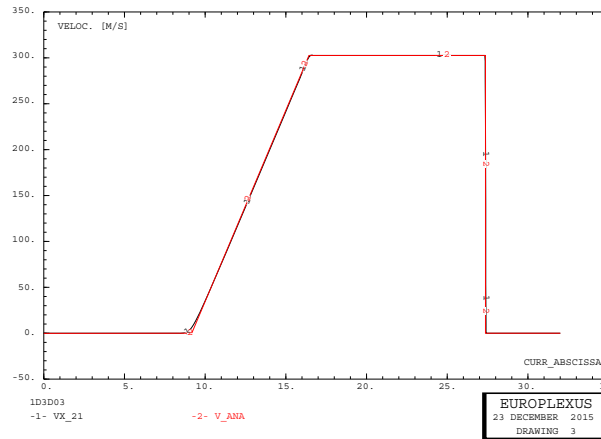
Table 3: Mixed-dimensional simulations.



(a) Pressure



(b) Density



(c) Velocity

Figure 6: Results of test 1D3D03.

### 3.3.1 1D3D14

The Cast3m input file is shortly commented below.

```

opti echo 1;
opti dime 3 elem cub8;
opti trac psc ftra '1d3d14_mesh.ps';
opti sauv form '1d3d14.msh';
*
lhp3d = 12.0;
lhp1d = 4.0;
llp1d = 6.0;
llp3d = 10.0;
nhp3d = 1200;
nhp1d = 400;
nlp1d = 600;
nlp3d = 1000;
h = lhp3d / nhp3d;
tol = h / 10;
*
p0 = 0 0 0;
p1 = 0 h 0;
p2 = 0 h h;
p3 = 0 0 h;
bashp3d = manu qua4 p0 p1 p2 p3;
hp3d = bashp3d volu tran nhp3d (lhp3d 0 0);
baslp3d = bashp3d plus ((lhp3d+lhp1d+llp1d) 0 0);
lp3d = baslp3d volu tran nlp3d (llp3d 0 0);
pid1 = p0 plus (lhp3d 0 0);
pid2 = pid1 plus (lhp1d 0 0);
pid3 = pid2 plus (llp1d 0 0);
hp1d = pid1 d nhp1d pid2;
lp1d = pid2 d nlp1d pid3;
hp = hp3d et hp1d;
lp = lp3d et lp1d;
flui = hp et lp;
*
* raccords 3d-1d
*
facehp = bashp3d plus (lhp3d 0 0);
pfacchp = chan poi1 facehp;

```

```

elim tol (pfacehp et hp3d);
facelp = baslp3d;
pfacehp = chan poi1 facelp;
rachp = manu supe (p1d1 et facehp);
raclp = manu supe (p1d3 et facelp);
*
mesh = flui et facehp et facelp et rachp et raclp;
*
tass mesh noop;
sauv form mesh;
trac cach qual mesh;
*
fin;

```

The 1D and 3D zones are meshed in the standard way. However, in order to “connect” the various zones in the EPX model, two “joints” (*raccords*) are needed. These are elements of the **SUPE** (superelement) type in Cast3m. Such elements are defined just by a cloud of nodes, and are named **rachp** and **raclp**, respectively, in the above example.

The EPX input file reads:

```

1D3D14
ECHO
!CONV win
CAST mesh
TRID EULE
DIME JONC 10 TERM ! Total n. of nodes in a TUBM junction

```

Note that it is necessary to dimension for the 1D-3D junctions. The **JONC** keyword must be specified, followed by the total number of nodes in **TUBM** junctions.

```

GEOM CUVF hp3d lp3d TUVF hp1d lp1d CL3D facehp facelp
TUBM rachp raclp TERM
COMP DIAM DROI 0.011283792 LECT hp1d lp1d TERM
RACC TUBM LECT rachp TERM
NTUB LECT p1d1 TERM DTUB 0.011283792
FACE LECT facehp TERM COEF 1.0
RACC TUBM LECT raclp TERM
NTUB LECT p1d3 TERM DTUB 0.011283792
FACE LECT facelp TERM COEF 1.0
! Attention: the TUBM elements (rachp and raclp) are NOT included
! in the "mesh" object (although they ARE indeed passed in from Cast3m).
! For this reason we must add them explicitly in the GROU directive below
! if we want to have them in the extracted element groups.
GROU 2 'nrachp' LECT mesh rachp raclp TERM
COND XB GT 11.99 COND XB LT 12.01
'nraclp' LECT mesh rachp raclp TERM
COND XB GT 21.99 COND XB LT 22.01
COUL ROUG LECT hp TERM
TURQ LECT lp TERM
VERT LECT rachp raclp TERM

```

The **TUBM** elements of EPX are associated with the junctions created in Cast3m (**rachp** and **raclp**). A diameter (**DIAM**) must be assigned to the 1D parts of the mesh, i.e. those meshed by the **TUVF** elements. The **DROI** keyword states that the tube has a constant cross-section.

The **RACC** directive must be used to define the geometric characteristics of the junctions. The **TUBM** keyword introduces the name of the Cast3m object associated with the junction element (**rachp** or **raclp** here), **NTUB** introduces the name of the 1D node to be connected to the 3D zone (**p1d1** and **p1d3**, respectively), **DTUB** allows to specify the internal diameter of the connected tube (here assumed to be 0.011283792 units, as specified above in the **DIAM** directive), **FACE** introduces the name of the Cast3m object specifying the (nodes of the) 3D face connected (**facehp** and **facelp**) and finally **COEF** is a scaling coefficient which is applied to the cross-section of the 3D mesh when computing fluxes across a joint (here we assume **COEF 1.0**). This may be used e.g. to exploit symmetries by modelling only part of a symmetric 3D mesh.

The choice of the **DIAM**, **DTUB** and **COEF** values deserves some explanation. The scope here is to join two different models (one in 3D and one in 1D) of the same physical tube, ideally straight and of constant cross-section. In order to avoid any spurious reflections of waves at the junctions, the (numerical) *cross sections* of the two models should be the same.

In the 3D part of the mesh, the (single) element forming the model is a regular hexahedron of  $L = 0.01$  m side. Therefore the effective cross-section of the 3D model is  $A = L^2 = 0.0001$  m<sup>2</sup>. In the 1D part, the cross section is  $S = \pi D^2/4$  where  $D$  is the diameter of the tube (assuming a circular cross-section). In order to have  $S = A$ , the diameter must be  $D = \sqrt{4A/\pi}$ , which in this case gives  $D = \sqrt{4 \cdot 0.0001/\pi} = 0.011283792$  m. In the above input, we have set the tube diameter (**COMP DIAM DROI**) to precisely this value. Then, in both **RACC** directives, the **DTUB** parametre is set to the

same value. In fact, DTUB can be interpreted as the diameter of a circular tube with a cross-section equivalent to the actual 3D cross-section.

Since the matching of the two models is already correct with these choices, we can simply set the COEF value to 1.0. Using a COEF different from 1.0 would only make sense in more complex situations, e.g. when joining a partially-meshed 3D part (exploiting symmetries, for example) with an equivalent 1D part.

Note that, as commented in the input file, a strange behaviour was observed when executing the test case. Although the two junction elements `racbp` and `raclp` created in Cast3m have been added (ET operator) to the global object `mesh` passed to EPX, and although they are recognized by the GEOM directive when the TUBM elements are associated with them, these elements are *not* drawn if one visualizes the entire mesh (i.e. the `mesh` object).

In order to make them appear in the drawings, we are obliged to re-declare them as shown in the input file above.

```
MATE GAZP RD 14.4 GAMM 1.4 CV 720 PINI 12.E5 PREF 1.E5
      LECT hp TERM
      GAZP RD 1.20 GAMM 1.4 CV 720 PINI 1.0E5 PREF 1.E5
      LECT lp TERM
! In order to obtain a printout at least of the 3D VFCCs I am obliged
! to use a different material for racbp and raclp, but with the
! same characteristics as the materials used for hp and lp, respectively
      GAZP RD 14.4 GAMM 1.4 CV 720 PINI 12.E5 PREF 1.E5
      LECT racbp TERM
      GAZP RD 1.20 GAMM 1.4 CV 720 PINI 1.0E5 PREF 1.E5
      LECT raclp TERM
ECRI ECRO VFCC TFRE 1.E-3
! NOPO NOEL
FICH ALIC TFRE 1.E-3
OPTI NOTE CSTA 0.75
LOG 1
      VFCC FCON 6 ! hllc solver
      ORDR 2 ! order in space
      UTPS 2 ! order in time
! RECO 1 ! Not accepted by CAL_VFCC_1D
CALC TINI 0 TEND 20.E-3
FIN
```

Another odd outcome of this calculation is that no output concerning the VFCCs is printed on the listing, despite the VFCC keyword being present in the ECRI directive.

Source code inspection reveals that this is likely a bug. At the moment, the code is unable to print the results for 1D VFCCs, but it is able to print those for 2D or 3D VFCCs. Since the results are printed material by material, when both 1D and multi-D VFCCs share the same material no results are printed at all! To circumvent this bug and to obtain a printout at least for the 3D part of the VFCCs, we are obliged to assign separate (but identical) materials to the junction elements, as shown in the input file above.

The solution of this problem is shown in Figure 7 and is in very good agreement with the analytical result.

### 3.3.2 1D3D15

This test is similar to test 1D3D14 but uses an alternative (but equivalent) definition of the DIAM, DTUB and COEF parameters, which should in principle give the same results in practice.

Here we choose a diameter  $D$  of the tube (DIAM DROI) of 0.01 m, i.e. equal to the side  $L$  of the 3D mesh, instead of 0.011283792 m. The value of the tube diameter in the mesh junction directive, (RACC DTUB) is set to the same value ( $D = 0.01$  m). The effective cross-section area of the 1D model is therefore  $S = \pi D^2/4$ , while the effective area of the 3D mesh is still  $A = L^2 = D^2$ . In order to connect the two models without spurious wave reflections the two numerical areas must be the same.

To this end, one may exploit the scaling factor  $\kappa$  (COEF), which EPX tacitly applies to the effective area of the 3D part. In other words, EPX multiplies the area of the 3D part by  $\kappa$  when computing transport terms at the junction. This is equivalent to stating that, in order to join two models with different effective cross sections, one must define COEF as the ratio between the 1D model effective area  $S$  and the 3D model effective area  $A$ . In other words,  $\kappa = S/A$ .

In the present case one has  $\kappa = S/A = \pi/4 = 0.78539816$ . The input related to junctions definition becomes therefore:

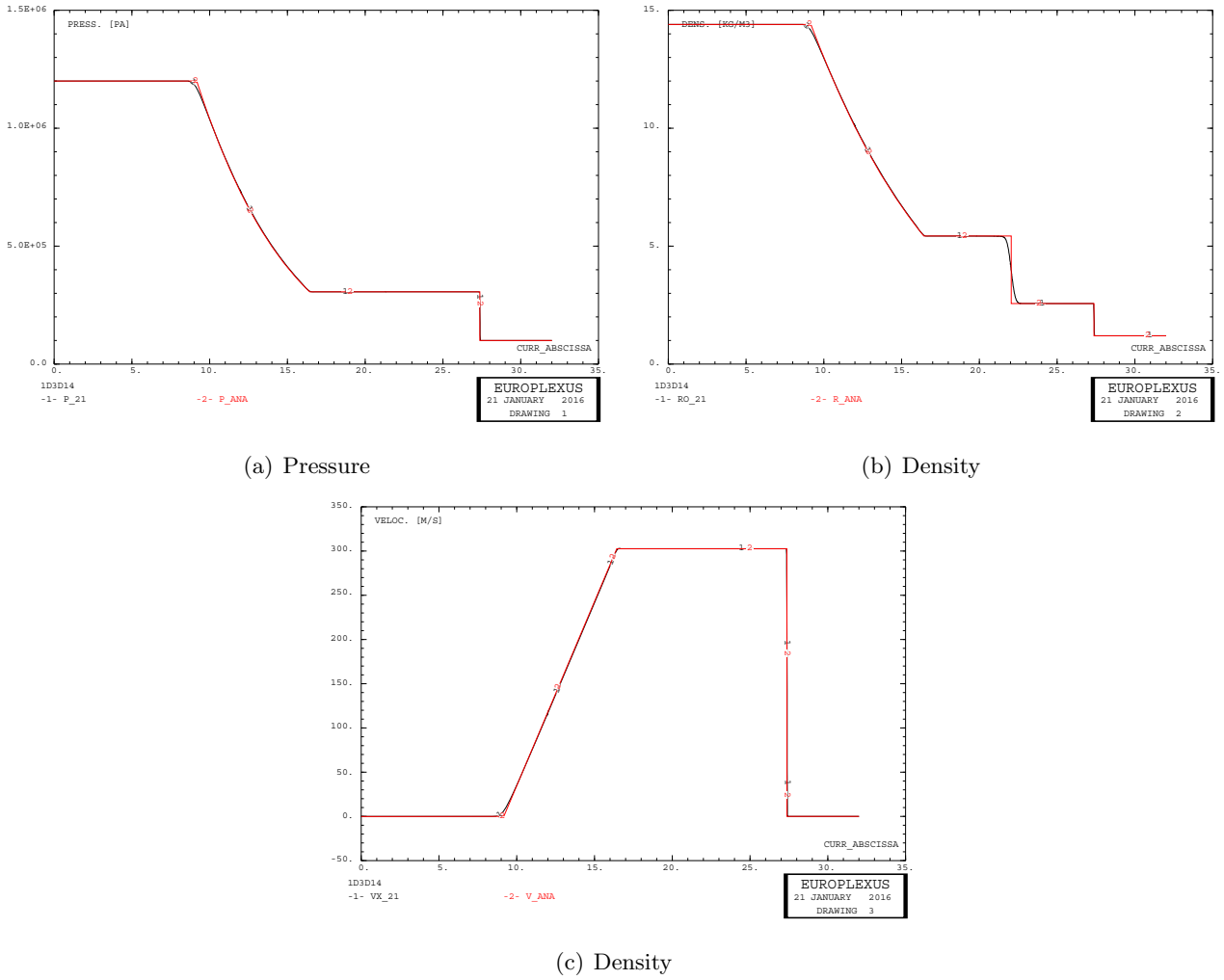


Figure 7: Results of test 1D3D14.

```

COMP DIAM DROI 0.01 LECT hpid lpid TERM
RACC TUBM LECT rachp TERM
NTUB LECT pid1 TERM DTUB 0.01
FACE LECT facehp TERM COEF 0.78539816
RACC TUBM LECT raclp TERM
NTUB LECT pid3 TERM DTUB 0.01
FACE LECT facelp TERM COEF 0.78539816

```

Results of this calculation are identical to those of test 1D3D14 and are not shown for brevity.

### 3.3.3 1D3D16

This test is similar to test 1D3D14 but uses the TUBE 1D fluid *finite element* instead of the TUVF 1D fluid finite volume. The 3D part of the mesh remains composed of CUVF finite volumes. Therefore, in this example, the fluid mesh is hybrid, being partly composed of finite volumes and partly of finite elements (and the two parts are connected together).

This might look a strange choice, but it was inspired by some benchmark tests present in the EPX standard suite of test cases. The only modification needed in the input file, besides of course replacing the TUVF by TUBE, is the necessity to add an explicit “link” on the junctions:

```

LIAI TUBM LECT rachp TERM
TUBM LECT raclp TERM

```

Note that the LIAI version of the link has been used, instead of the LINK COUP version, simply by similarity with the benchmark example. It is not known whether this type of link has been already ported to the new links model (LINK).

The results obtained are shown in Figure 8. The solution is stable, but results are not as accurate as those of the purely Finite Volume discretization.

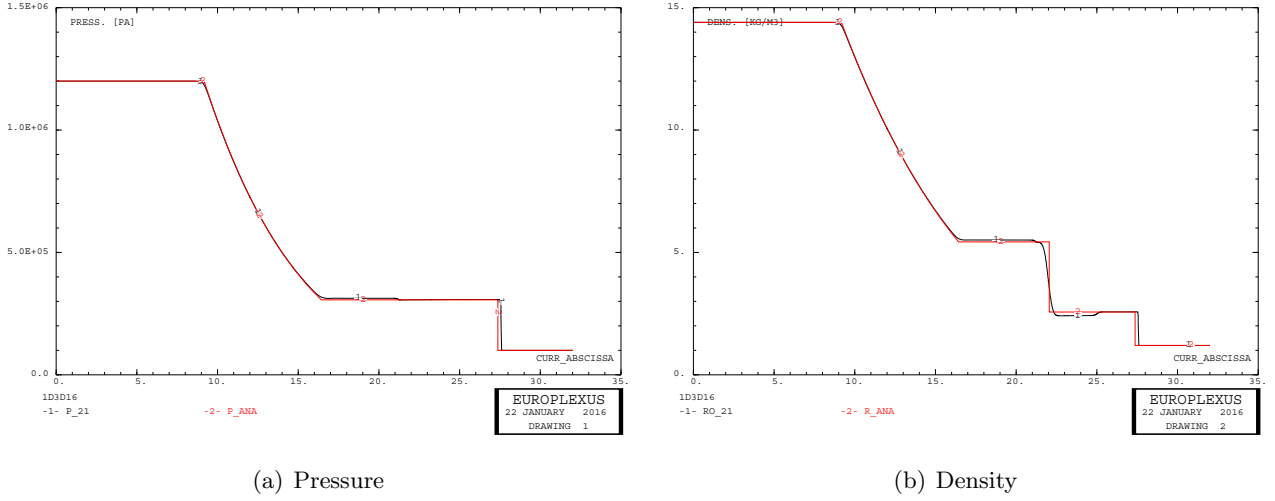


Figure 8: Results of test 1D3D16.

### 3.3.4 1D3D17

This test is similar to test 1D3D16 but an attempt is made to ameliorate the results by activating full second-order formulation in space, i.e. by activating the `RECO 1` option. Note that this is now possible since no 1D cell-centered finite volumes are present in the mesh.

This calculation is considerably more expensive than the previous one in terms of CPU time, see Table 3. The results obtained are shown in Figure 9 but, unfortunately, they are as inaccurate as those obtained previously.

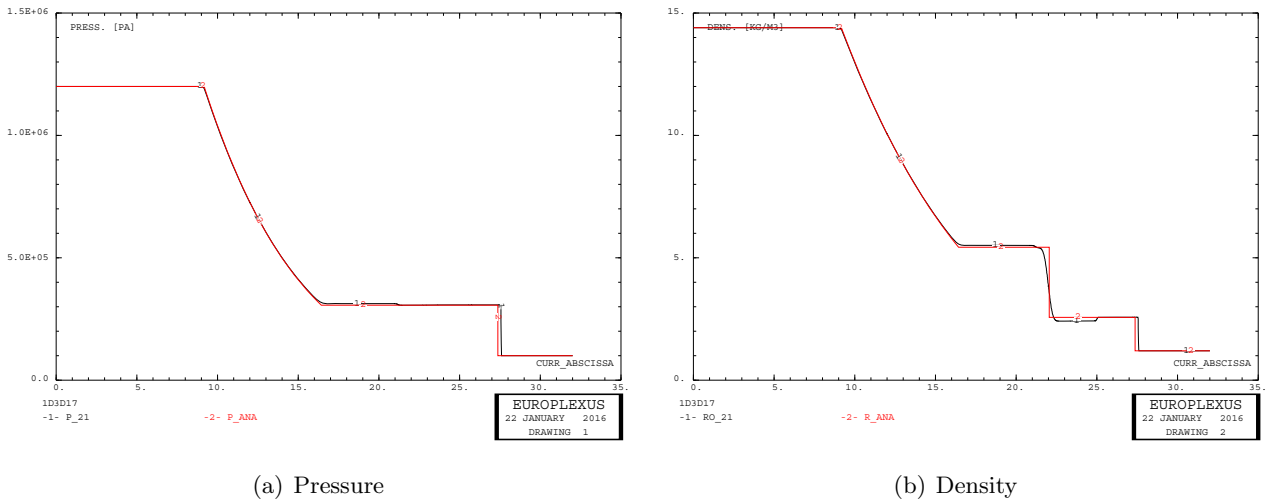


Figure 9: Results of test 1D3D17.

## 3.4 Application to NTNU shock tube

The techniques described in the previous Sections are now tentatively applied to the simulation of the NTNU shock tube facility. The scope is to reduce the CPU time needed, especially in view of many repetitive simulations (parametric studies), without sacrificing the accuracy.

The phase of the simulation where one can hope to gain CPU time is the initial one, when the pressure shock wave travels along the tube and before it reaches the test section of the device. With typical values of initial pressure in the driver section of the facility of 50 bar, the time needed for the shock wave to reach the test zone is estimated to be about 20.5 ms. This by assuming that the test zone starts about 1 m upstream of the test specimen. A simple square thin plate is taken

as a specimen in these simulations, since the scope here is to model the preliminary phases of the experiment and not the real behaviour of the specimen.

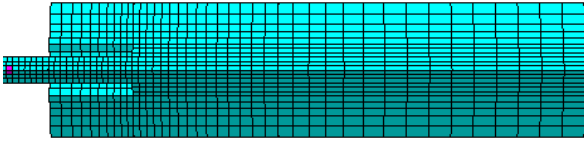
The basic model (coarse mesh) of the test rig is shown in Figure 10. By exploiting the two symmetry planes, only 1/4 of the rig is modelled.

SHOC51  
TIME: 0.00000E+00 STEP: 0



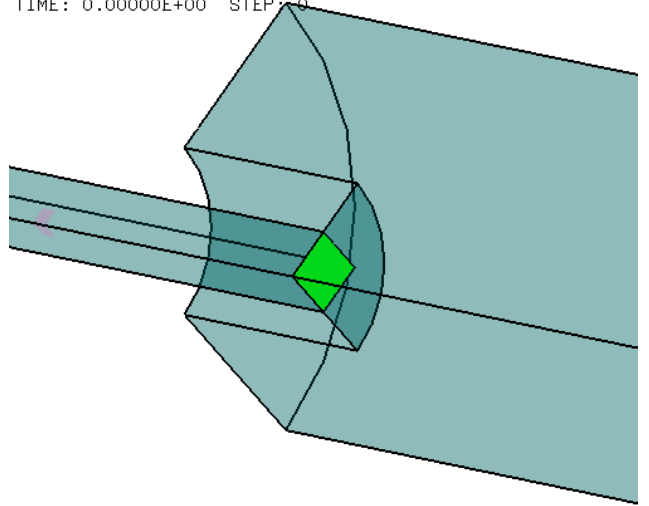
(a) Complete rig

SHOC51  
TIME: 0.00000E+00 STEP: 0



(b) Test section

SHOC51  
TIME: 0.00000E+00 STEP: 0



(c) Specimen (in green)

Figure 10: Basic model of the NTNU test rig.

The calculations done are summarized in Table 4.

Case	1D-3D	Adaptivity	Trigger [s]	@step	@CPU [s]	Total CPU [s]	NOCR	TBL0
SHOC51	No	No	—	(4380)	(118)	132	No	No
SHOC52	No	Yes	—	(4380)	3370	3868	No	No
SHOC53	No	Yes	20.51	4380	452	997	No	No
SHOC54	No	Yes	—	—	—	—	Yes	No
SHOC55	No	Yes	20.58	769	81	605	Yes	Yes
SHOC61	Yes	No	—	(4380)	(30)	33	No	No
SHOC62	Yes	Yes	—	(4380)	2272	2611	No	No
SHOC65	Yes	Yes	20.43	759	72	457	Yes	Yes

Table 4: Shock tube simulations.

### 3.4.1 SHOC51

This calculation is purely 3D and uses the coarse (base) mesh shown in Figure 10, which consists of 5,286 elements CUVF for the fluid, 9 elements Q4GS for the plate and 60 elements CL3D for the



absorbing boundary zone on part of the final tank downstream the plate. The mesh size is 0.05 m both in the fluid and in the structure. This mesh is far too coarse for an accurate solution, but the scope here is just to set a reference for the CPU time needed.

The calculation is performed up to  $t_f = 23$  ms. It takes 4,380 time steps and 118 s of CPU time for the shock wave to arrive at a distance of 1 m upstream the specimen, at the expected time of 20.5 ms. In the following, we will denote this time interval as the *pre-loading phase*. One element of the plate fails and gets eroded at step 4,885 and time 22.9 ms.

Figure 11 shows the pressure field in the test zone and the deformed, partially eroded plate at the end of the simulation.

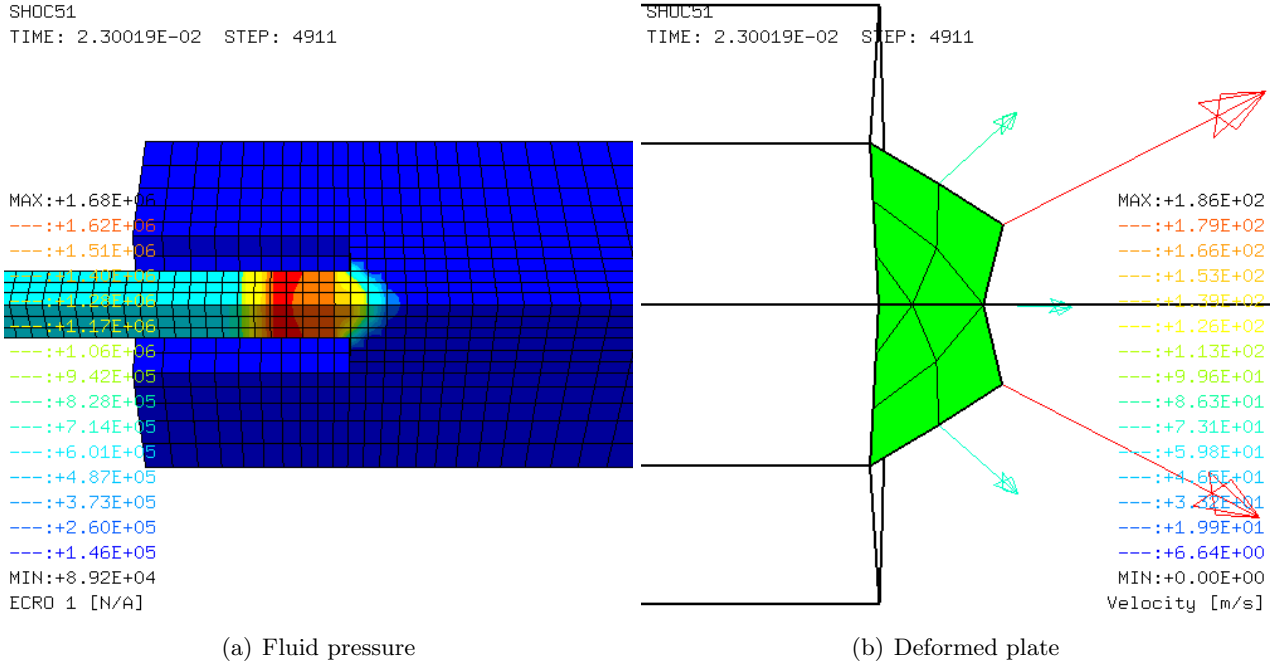


Figure 11: Some results of test SHOC51.

Figure 12 shows the pressure and the fluid velocity profiles along the axis of the test rig at  $t = 20$  ms (red curves) and at  $t = 23$  ms (black curves). The first time is slightly before the arrival of the shock wave in the test zone, the second time is the final time of the simulation. One may observe the partial pressure reflection at the plate and the passage of fluid across the failing plate.

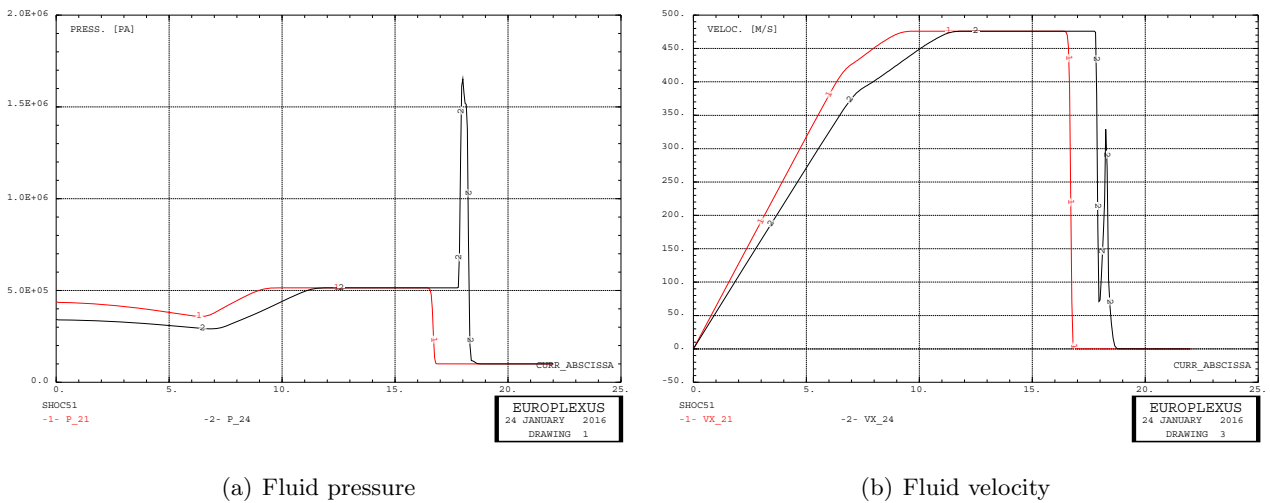


Figure 12: Further results of test SHOC51.

### 3.4.2 SHOC52

This test case is similar to the previous one but the fluid mesh is adaptively refined in the test zone near the plate by means of the FLSW ADAP directive, up to a maximum level of 4 (LMAX 4). A total number of around 125,000 fluid elements (of which about 5,000 base elements and 120,000 descendents) are needed. This type of mesh adaptivity is called FSI-driven adaptivity.

Two input modifications are needed with respect to the previous case. First, some dimensioning must be prescribed for the mesh adaptivity:

```
DIME ADAP NPOI 115000 CUVF 120000 NVFI 380000 ENDA
... TERM
```

Then, the ADAP sub-directive must be added to the FLSW command:

```
LINK DECO FLSW ...
ADAP LMAX 4 SCAL 16
```

The plate is not refined. Of course, this is unrealistic but is appropriate for the scope of the present investigation. During the calculation, the refined fluid mesh “follows” the motion of the plate and therefore the mesh connectivity is not static.

This calculation is much more expensive than the previous one, requiring 4,380 time steps and 3,370 s of CPU time for the pre-loading phase. Six plate elements fail and are eroded, between 22.7 and 23 ms.

Note that the number of time steps in the pre-loading phase is the same as in the previous case, despite the fact that here a much finer fluid mesh is present in the test zone. This is due to the fact that in this model the most critical time step occurs in the plate elements (which are the same in both cases), so that the time discretization is driven by the structure, not by the fluid. The difference in CPU between cases SHOC51 and SHOC52 is therefore due basically to the number of fluid elements, which is much larger in the second model.

Figure 13 shows the initial refined mesh and the final pressure field in the test zone.

```
SHOC52
TIME: 0.00000E+00 STEP: 0
```

```
SHOC52
TIME: 2.30009E-02 STEP: 5039
```

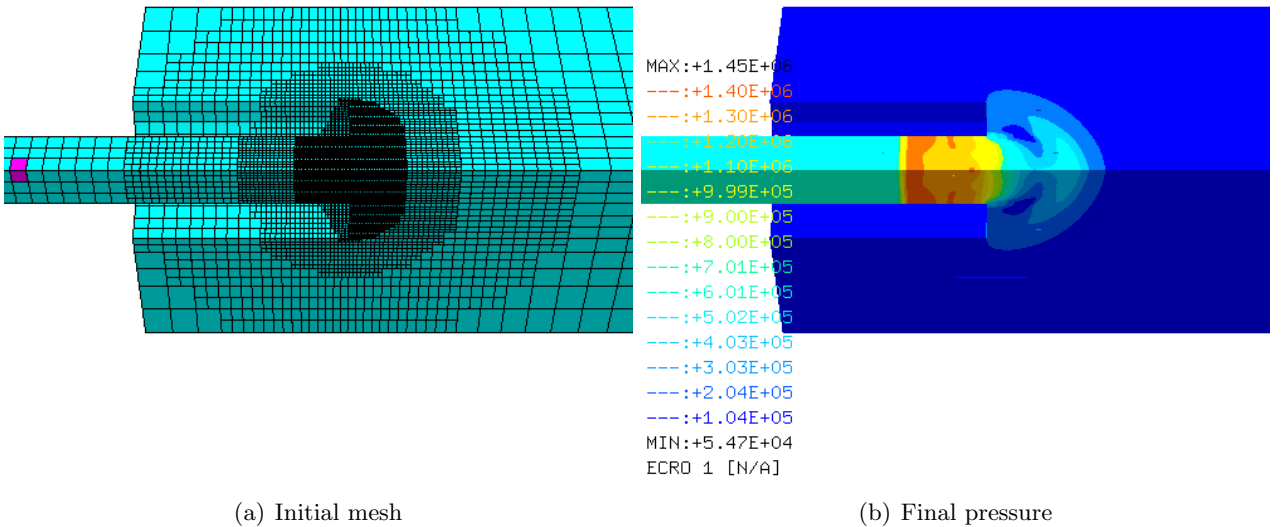


Figure 13: Some results of test SHOC52.

Figure 14 shows the pressure and the fluid velocity profiles along the axis of the test rig at  $t = 20$  ms (red curves) and at  $t = 23$  ms (black curves).

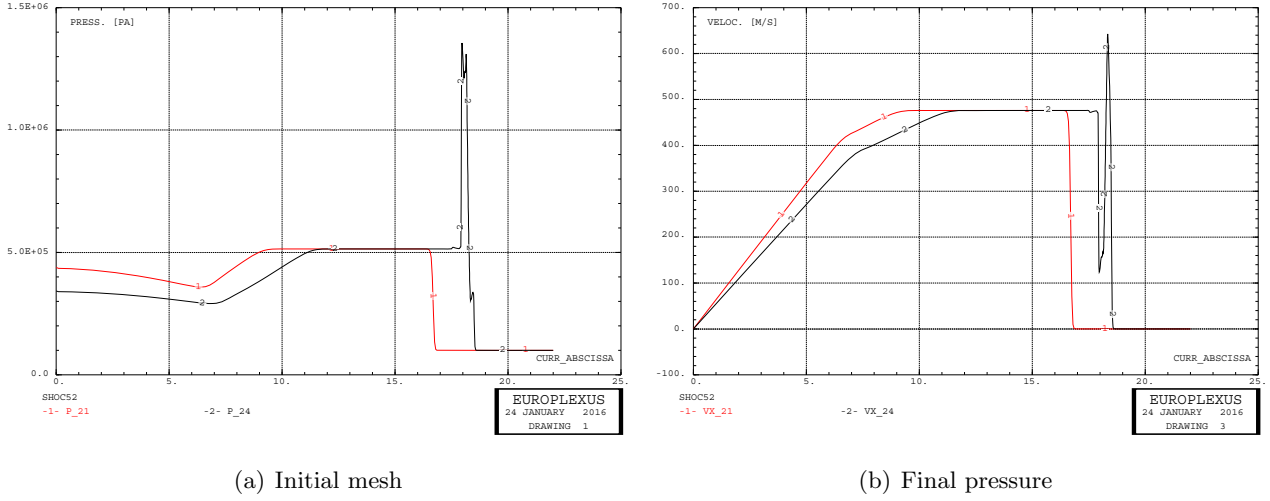


Figure 14: Further results of test SHOC52.

### 3.4.3 SHOC53 – Triggering the mesh adaptation

Before passing to mixed-mesh solutions, we exploit a few other means allowing to reduce the CPU time in this type of numerical simulations. The first observation is that it is useless to refine the fluid mesh in the test zone from the very beginning of the calculation. This operation is only needed when the pressure wave is about to hit the plate.

To achieve this we use a “trigger”, i.e. we monitor a quantity (e.g. the pressure) at a chosen location (e.g. 1 m upstream the specimen) and we fire up the mesh adaptation as soon as the monitored quantity reaches a certain value. In EPX, the option `ADAP TRIG` can be used for this purpose.

In the EPX input file, first we choose the trigger position:

```
COMP GROU 3 ...
      'trigger' LECT tube TERM COND NEAR POIN -1.01 0 0
      ...
```

The **trigger** object is defined as a group of elements, in this case the (single) element whose centroid is closest to the point specified, located at 1.01 m upstream the plate (which is placed in the origin of the global reference system). This element is shown in red in some of the preceding Figures.

Then in the options section of the input file it is sufficient to add:

```
OPTI ...
      ADAP TRIG ECRO 1 TVAL 1.02E5 LECT trigger TERM
```

The monitored quantity is `ECRO` component 1, which for the fluid material is the (absolute) pressure, and the chosen threshold value is 1.02 bar. Since the initial pressure in the tube is 1 bar, this means that the trigger will be activated (i.e. the mesh will be first refined) as soon as the pressure in the trigger element increases by 2%.

As it could be expected, in this calculation the trigger is activated at 20.5 ms, step 4,380. The CPU time needed to reach that point of the calculation (pre-loading phase) is 452 s.

Figure 15 shows the initial (coarse) mesh and the final pressure field in the test zone. The latter is practically identical to the one of case SHOC52, see Figure 13.

Figure 16 shows the pressure and the fluid velocity profiles along the axis of the test rig at  $t = 20$  ms (red curves) and at  $t = 23$  ms (black curves). These results are practically identical to those of case SHOC52, see Figure 14.

### 3.4.4 SHOC54

Although the previous solution SHOC53 was almost 7.5 times faster than the reference fine-mesh solution SHOC52 in the pre-loading phase, it remains that the time step used in this phase is still the

SHOC53  
TIME: 0.00000E+00 STEP: 0

SHOC53  
TIME: 2.30009E-02 STEP: 5039

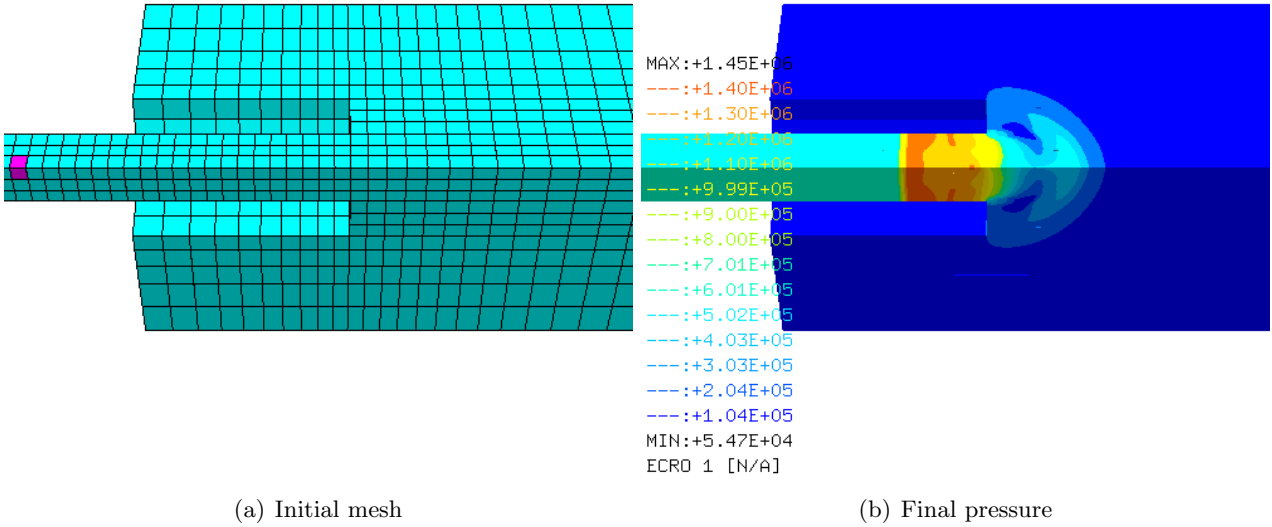


Figure 15: Some results of test SHOC53.

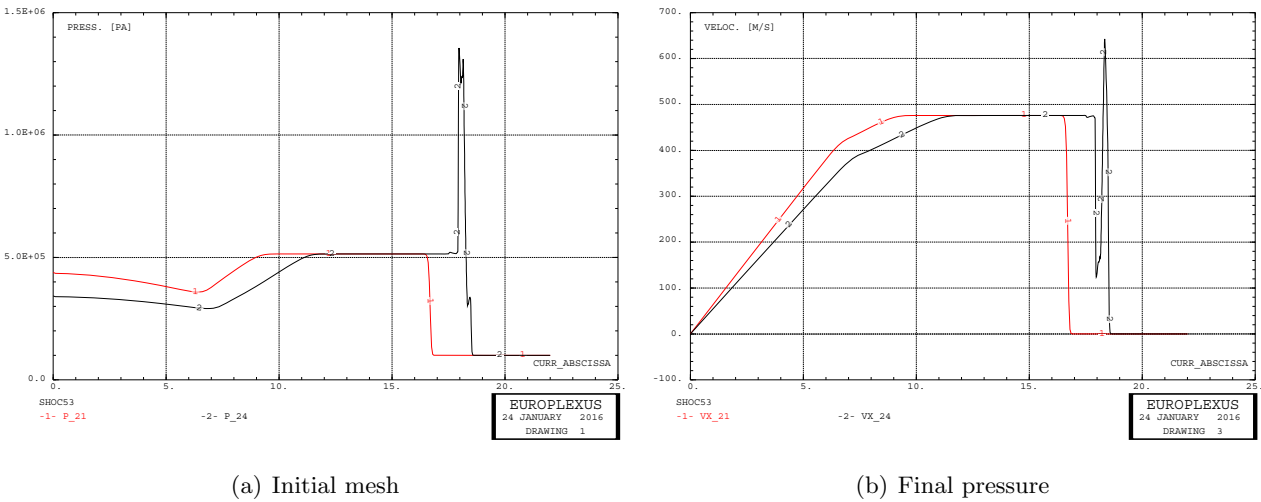


Figure 16: Further results of test SHOC53.

one dictated by the structure, and therefore far too small for the fluid domain. Since the structure is not defined in this phase, one might expect being able to adopt a much larger time step, suitable for the fluid alone.

This calculation is similar to SHOC53 but we add the following option:

```
OPTI ...  
NOCR TRIG LECT plat TERM
```

This tells EPX to skip critical step calculation `NOCR` for the elements of the `plat` object (i.e. all structural elements present in the model), until the adaptivity trigger is fired (`TRIG`).

This causes the initial time steps in this calculation to be 10 times larger than in the previous one, potentially a large saving in CPU time.

However, the calculation results are not the expected ones. Several “spurious” element failures and erosions occur in the plate as early as 1.3 to 1.7 ms, which is totally unphysical since the pressure wave has not yet hit the plate at that time.

The trigger is fired (correctly) at 20.5 ms (step 769) but immediately after the calculation fails

due to inverse mapping in the fluid hexahedra (this problem is probably related to excessive distortion and weird shape of some plate elements.)

Visualization of intermediate results reveals that the plate undergoes some relatively large unphysical motions and deformations during the early pre-loading phase. These are thought to be due to numerical instabilities. Even tiny out-of-equilibrium forces acting on the plate (probably due to the FLSW model of fluid-structure interaction) are probably amplified by the lack of stability of the explicit time scheme (the Courant number is of the order of 10 for the shell elements in this phase) and result in a macroscopic instability in the structural model.

### 3.4.5 SHOC55

In order to avoid the instability observed in the previous calculation during the pre-loading phase, an obvious remedy is to completely block the structure until the trigger is fired up.

To this end, the `TBLO TRIG` directive is added, so that the coupled links directive results as follows:

```
LINK COUP
BLOQ 123456 LECT c2s c3s TERM
BLOQ 345 LECT c1s DIFF p0s p1s TERM
BLOQ 246 LECT c4s DIFF p0s p3s TERM
BLOQ 23456 LECT p0s TERM
TBLO 126 TRIG LECT c1s DIFF p0s p1s TERM
135 TRIG LECT c4s DIFF p0s p3s TERM
1 TRIG LECT p0s TERM
123456 TRIG LECT plat DIFF c1s c2s c3s c4s TERM
```

The `TBLO TRIG` directive defines some time-limited blockages which expire as soon as the adaptivity trigger is fired up. Note that we time-block all the degrees of freedom of the plate which are *not* subjected to a permanent blockage (`BLOQ`).

This calculation takes 769 steps to reach the end of the pre-loading phase (trigger), which occurs at 20.6 ms. The CPU time for that phase passes from 452 to only 81 s and so is even faster than the coarse-mesh simulation SHOC51 (because of the longer time steps).

Figure 17 shows the initial (coarse) mesh and the final pressure field in the test zone. The latter is practically identical to the cases SHOC52 and SHOC53, see Figures 13 and 15.

SHOC55  
TIME: 0.00000E+00 STEP: 0

SHOC55  
TIME: 2.30017E-02 STEP: 1410

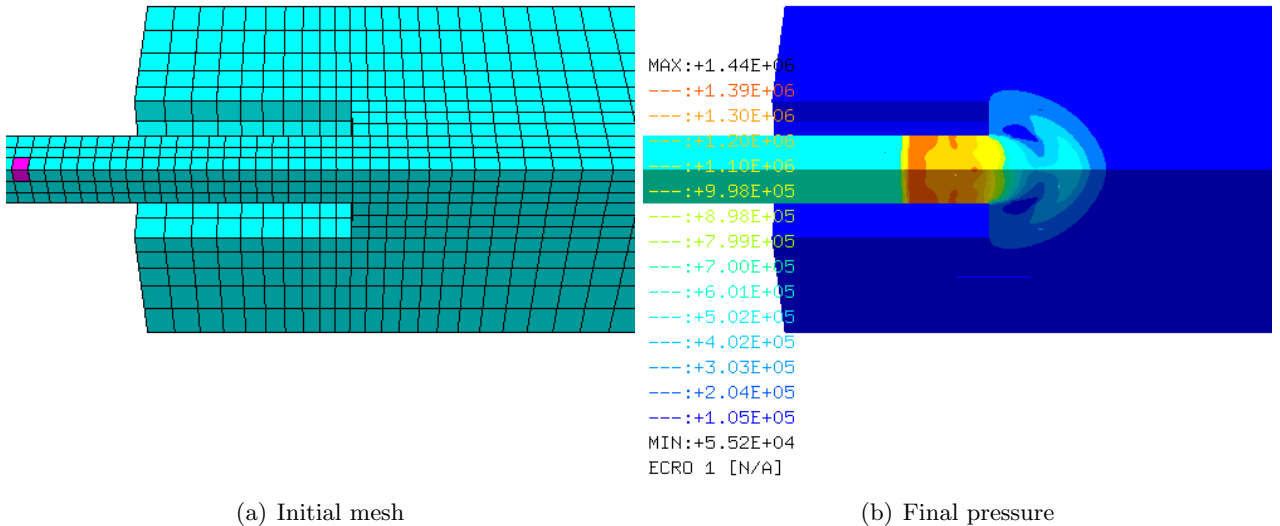
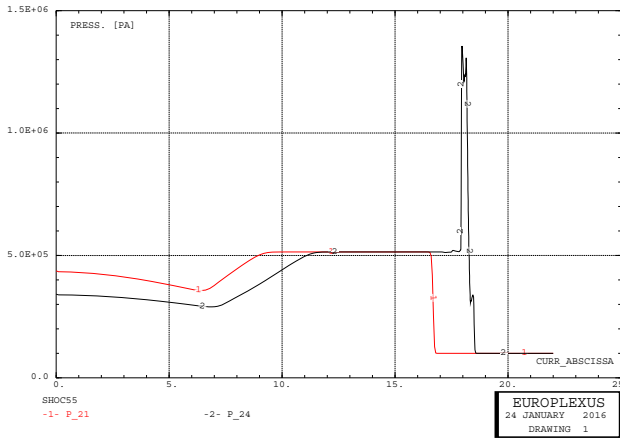
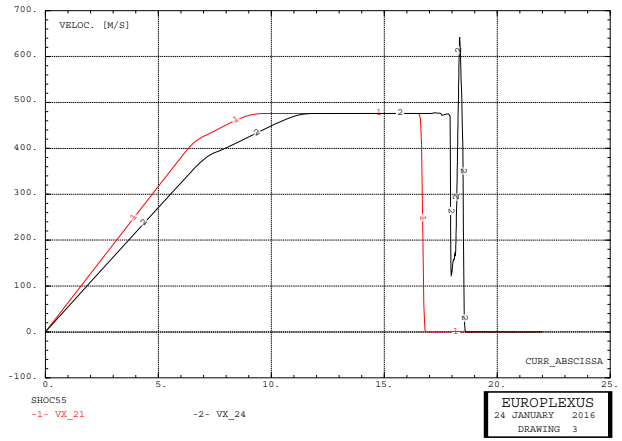


Figure 17: Some results of test SHOC55.

Figure 18 shows the pressure and the fluid velocity profiles along the axis of the test rig at  $t = 20$  ms (red curves) and at  $t = 23$  ms (black curves). These results are practically identical to cases SHOC52 and SHOC53, see Figures 14 and 16.



(a) Initial mesh



(b) Final pressure

Figure 18: Further results of test SHOC55.

### 3.4.6 SHOC61

We now test out 1D-3D mixed models of the NTNU shock tube. This first test is similar to case SHOC51, i.e. it uses a basic, unrefined mesh all over, but the initial part of the shock tube, up to a distance of 1.2 m from the specimen (test plate) is meshed by 1D finite volumes TUVF instead of 3D finite volumes CUVF.

The mixed mesh in the vicinity of the test section is shown in Figure 19. The small black spheres are the nodes of the 1D-3D junction element (TUBM). The magenta cube marks the position of the trigger (1 m upstream of the plate), but this is unused in the present calculation.

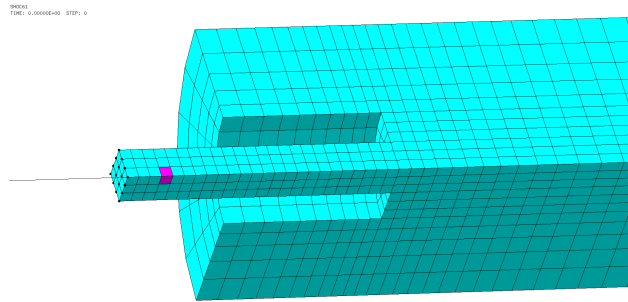


Figure 19: Mixed 1D-3D mesh of the test rig for case SHOC61.

The 1D part of the mixed mesh model is built in Cast3m by the following commands:

```

. . .
pid1 = (0 - ltube) 0 0;
pid2 = pid1 plus (ltubhp 0 0);
tubehp = pid1 d pid2 dini d1 dfin d1;
lenlp3d = 1.2;
pid3 = (0 - lenlp3d) 0 0;
tubelp1 = pid2 d pid3 dini d1 dfin d1;
stub3d = sflu1 plus pid3;
tubelp3 = stub3d volu dini d1 dfin d1 tran (lenlp3d 0 0);
tubelp = tubelp1 et tubelp3;
*
rac1p = manu supe (pid3 et stub3d);
*
tube = tubehp et tubelp;
*
flui = tube et tank;
elim tol flui;
. . .
*
mesh = flui et plat et abso et rac1p;
tass mesh noop;
sauv form mesh;
*
fin;

```

Note the construction of the junction element (**rac1p**) by the MANU SUPE directive. The **p1d3** node is the 1D node belonging to the junction while the **stub3d** surface (made of quadrilateral elements) is the 3D surface belonging to the junction. These will then be needed in EPX to define the junction by means of the RACC TUBM directive.

The EPX input file reads (only the interesting parts are shown below):

```

. . .
DIME JONC 17 ! Total n. of nodes in a TUBM junction
NALE 1 NBLE 1 TERM
GEOM CUVF tubelp3 tank TUVF tubehp tubelp1 Q4GS plat
! Attention: we are obliged to assign (dummy) CL3D elements to the stub3d
! object (3D part of the 1D-3D junction), because the name of this object
! is needed below in the RACC TUBM FACE directive and it seems that the
! code expects there an object made of elements, not just of nodes
! (although only the nodes are really needed, at least according
! to the manual)
CL3D abso stub3d
TUBM rac1p TERM

```

We need to dimension for the junctions (DIME JONC). Here the number of nodes involved is 17 (1 node in the 1D part and  $4 \times 4 = 16$  nodes in the 3D part.) Note that, as usual, any TUBM elements must be declared in the GEOM directive *after* any CLxx elements, and these must be declared after all “regular” elements.

As mentioned in the comments of the input file, it seems that one is obliged to assign (dummy) CL3D elements to the **stub3d** object (i.e. to the 3D part of the 1D-3D junction), because the name of this object is needed below in the RACC TUBM FACE directive and it seems that the code expects there an object made of *elements*, not just of nodes (although only the nodes are really needed, at least according to the users’ manual.) These CL3D elements are dummy in the sense that they are not really used in the calculation, other than in order to define the **FACE** of the RACC TUBM. Note that, in particular, *no material* needs to be assigned to these dummy elements.

```

COMP DIAM DROI 0.1692568 LECT tubehp tubelp1 TERM
RACC TUBM LECT rac1p TERM
NTUB LECT p1d3 TERM DTUB 0.1692568
FACE LECT stub3d TERM COEF 1.0
EPAI 0.8E-3 LECT plat TERM
! Attention: the TUBM element (rac1p) is NOT included
! in the "mesh" object (although it IS indeed passed in from Cast3m).
! For this reason we must add it explicitly in the GROU directive below
! if we want to have it in the extracted element groups.
GROU 4 'endtube' LECT tube TERM COND XB GT -0.6
      'trigger' LECT tube TERM COND NEAR POIN -1.01 0 0
      'frefine' LECT flui TERM COND XB GT -1.2
              COND XB LT 1.0
      'nrac1p' LECT mesh rac1p TERM
              COND XB GT -1.201 COND XB LT -1.199
COUL ROUG LECT tubehp TERM
TURQ LECT tubelp tank DIFF trigger TERM
VERT LECT plat TERM
ROSE LECT trigger TERM
JAUN LECT abso TERM
BLEU LECT rac1p TERM

```

The computational diameter  $D$  of the 1D tube (DIAM) is evaluated by equating the tube’s cross-section  $S = \pi D^2/4$  to the cross-section of the 3D model of the tube  $A = L^2$ , where  $L = 0.15$  m is the side of the 3D model representing just 1/4 of the tube (thanks to symmetries). One obtains thus  $D = \sqrt{4A/\pi} = 0.1692568$  m.

```

GRIL LAGR LECT plat TERM
EULE LECT flui TERM
MATE GAZP RO 59.459 GAMM 1.4 CV 719.286 PINI 50.E5 PREF 1.E5
LECT tubehp TERM
GAZP RO 1.189 GAMM 1.4 CV 719.286 PINI 1.E5 PREF 1.E5
LECT tubelp1 rac1p TERM
! In order to obtain a printout at least of the 3D VFCCs I am obliged
! to use a different material for tubelp3 and other 3D parts, but with the
! same characteristics as the material used for tubelp1
GAZP RO 1.189 GAMM 1.4 CV 719.286 PINI 1.E5 PREF 1.E5
LECT tubelp3 tank TERM
VPJC RO 2700.0 YOUN 70.0E9 NU 0.3 ELAS 97.6E6 MXIT 200
QR1 13.2E6 CR1 1458.0 QR2 10.6E6 CR2 185.9
PDOT 5.E-4 C 1.4E-2 TQ 0.9 CP 910.0
TM 893.0 M 1.0 DC 1.0 WC 44.6E6
LECT plat TERM
CLVF ABSO RO 1.20
LECT abso TERM
. . .
OPTI . . .
VFCC FCON 6 ! hllc solver
ORDR 2 ! order in space
OTPS 2 ! order in time
RECO 1 ! Not accepted by CAL_VFCC_1D
CALC TINI 0 TEND 23.E-3
FIN

```

As in the previous mixed-mesh examples, more materials than should be strictly necessary need to be defined to circumvent the printout bug. Finally, full second order in space cannot be activated (RECO 1 is commented out) due to the presence of 1D VFCCs.

Figure 20 shows the pressure field in the test zone and the deformed, partially eroded plate at the end of the simulation, showing excellent agreement with the same results of case SHOC51, see Figure 11.

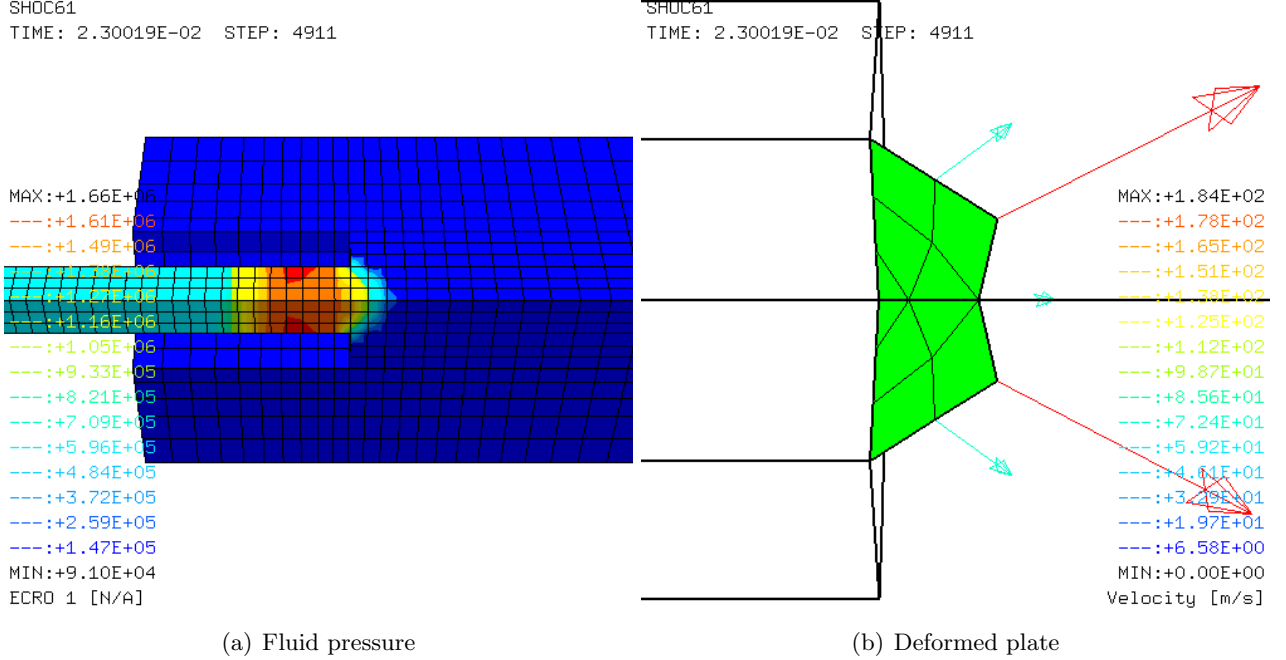


Figure 20: Some results of test SHOC61.

Figure 21 shows the pressure and the fluid velocity profiles along the axis of the test rig at  $t = 20$  ms (red curves) and at  $t = 23$  ms (black curves). Again, comparison with test SHOC51 is very good, see Figure 12, if one considers that the present solution is just first order in space while the previous one was second-order in space.

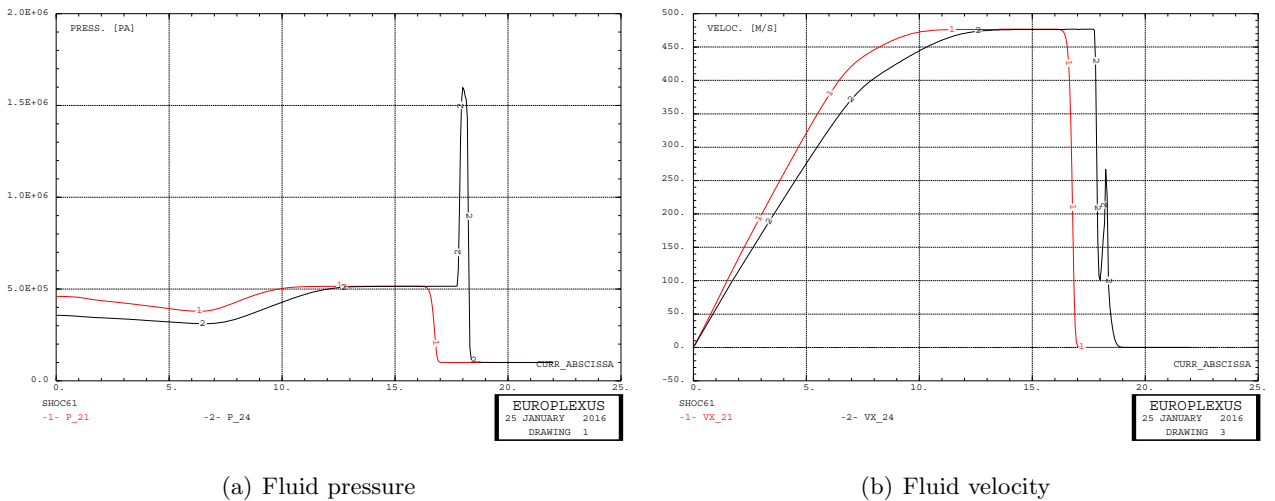


Figure 21: Further results of test SHOC51.

This calculation takes 33 s in the pre-loading phase and is almost 4 times faster than the corresponding purely-3D model, see Table 4. Of course, part of this speed-up is due to the fact that this is a first-order in space model instead of a second-order model.

The calculation SHOC51 was re-run as first-order in space (case SHOC50) by commenting out the



RECO 1 option, and it was found that the calculation time in the pre-loading phase was 67 s. So the use of a mixed 1D-3D model allows more than a factor 2 speed-up. Furthermore, the results of case SHOC50 were nearly identical to those of case SHOC61, indicating that the small differences with respect to solution SHOC51 are due to the order in space of the formulation, and not to the use of a mixed-dimensional model. The results of case SHOC50 are not shown for brevity, but the input files for this case are listed in the Appendix.

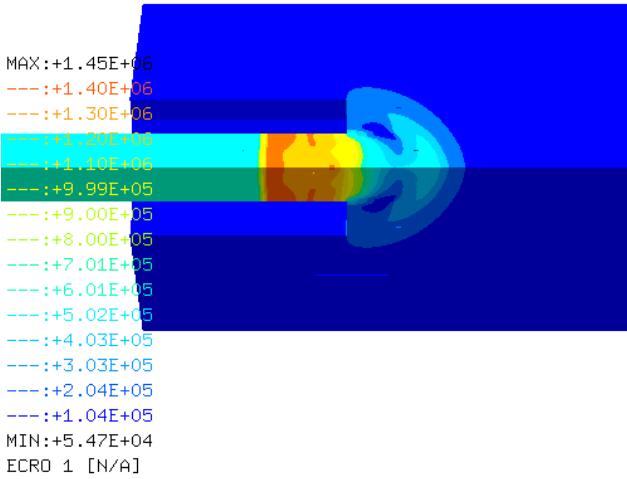
### 3.4.7 SHOC62

This model is similar to that of case SHOC52, i.e. the fluid mesh is adaptively refined near the plate from step 0, but uses a mixed 1D-3D fluid mesh like case SHOC61 (and first rather than second order in space). The CPU time for the pre-loading phase passes from 3,370 s to 2,272 s, a speed-up of almost 1.5.

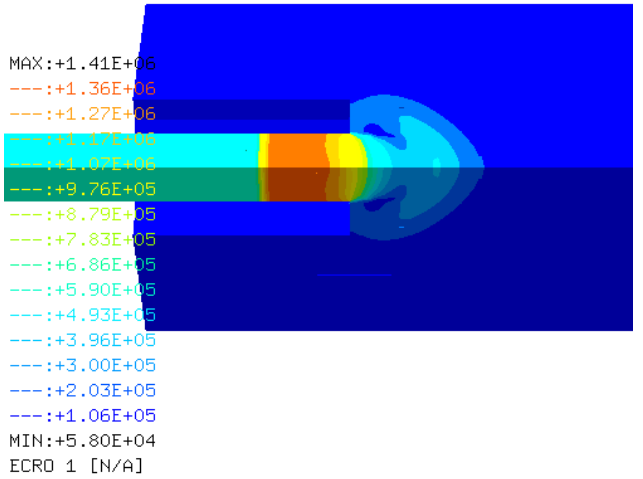
Figure 22 compares the final pressure field in the two solutions. Some slight differences can be observed (by the way, the color scales are automatic and do not use exactly the same values) but, again, part of this (modest) discrepancy is due to the different order in space of the two solutions.

SHOC52  
TIME: 2.30009E-02 STEP: 5039

SHOC62  
TIME: 2.30021E-02 STEP: 5029



(a) Case SHOC52



(b) Case SHOC62

Figure 22: Comparison of final pressures in cases SHOC52 and SHOC62

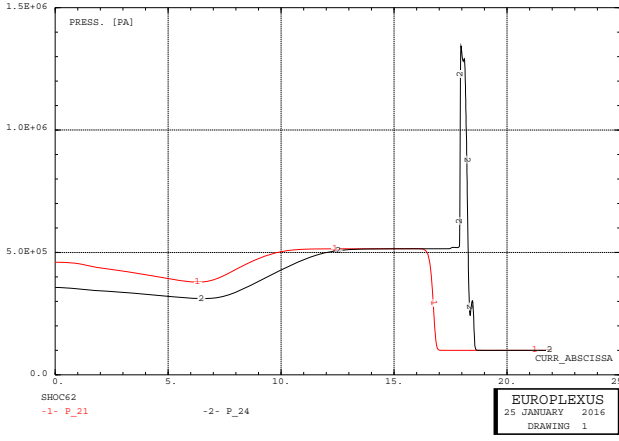
Figure 23 shows the pressure and the fluid velocity profiles along the axis of the test rig at  $t = 20$  ms (red curves) and at  $t = 23$  ms (black curves). Comparison with test SHOC52 (see Figure 14) is good if one considers the different order of the solutions.

### 3.4.8 SHOC65

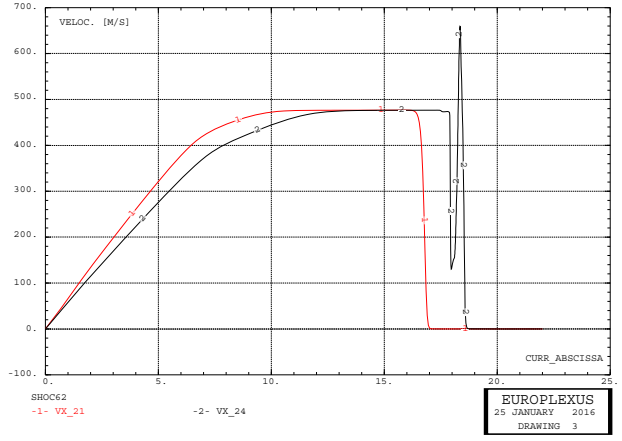
This model is similar to that of case SHOC55, i.e. a trigger is set to pilot the fluid mesh refinement, the plate elements stability and the plate element blockages, but uses a mixed 1D-3D fluid mesh like cases SHOC61 and SHOC62 (and a first-order in space formulation). The trigger fires up at step 759,  $t = 20.43$  ms and 71.6 s of CPU.

Figure 24 compares the final pressure field in solutions SHOC62 and SHOC65, showing very good agreement (both of these are first-order solutions.)

Figure 25 shows the pressure and the fluid velocity profiles along the axis of the test rig at  $t = 20$  ms (red curves) and at  $t = 23$  ms (black curves). Comparison with test SHOC62 (see Figure 23) is very good.



(a) Fluid pressure

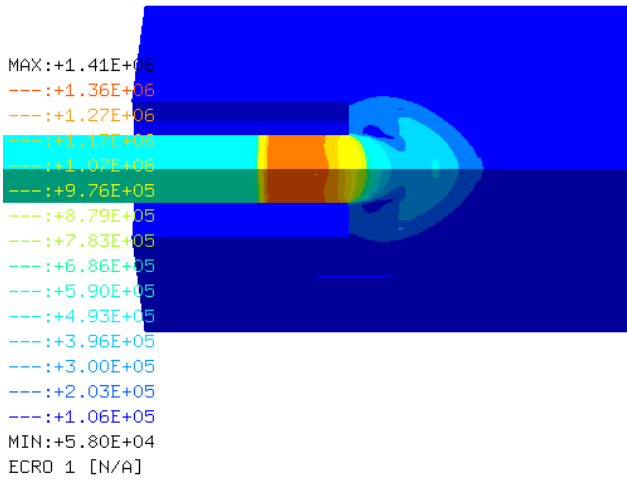


(b) Fluid velocity

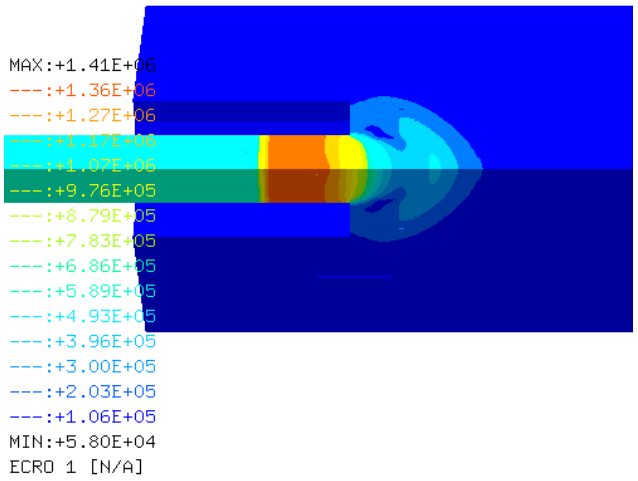
Figure 23: Further results of test SHOC62.

SHOC62  
TIME: 2.30021E-02 STEP: 5029

SHOC65  
TIME: 2.30002E-02 STEP: 1421



(a) Case SHOC62



(b) Case SHOC65

Figure 24: Comparison of final pressures in cases SHOC62 and SHOC65

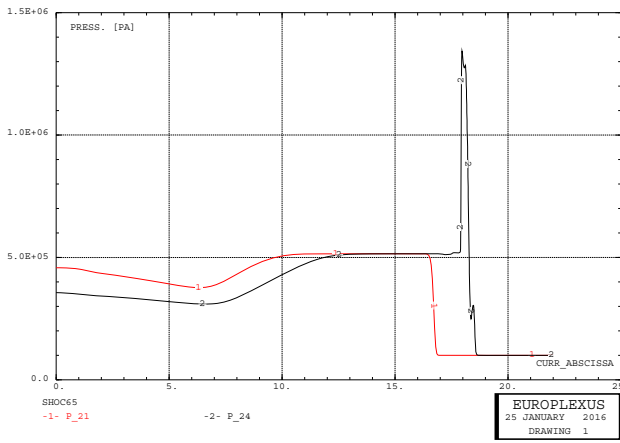
Compared with the analogous purely-3D solution (case SHOC55) we see that the speed up is relatively modest (1.13). However, the saving with respect to a “naive” solution such as case SHOC52 is very important (a factor of 47), showing that it is worthwhile to apply the techniques proposed in the present report especially when performing repetitive parametric studies of the behaviour of the shock tube test rig.

## 4 Conclusions and future work

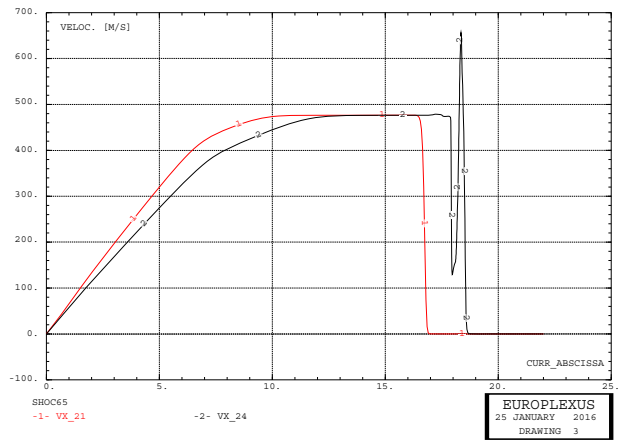
As shown in the previous Sections, coupling between 1D and 3D (or 2D) VFCC meshes can be exploited in EPX (together with other “tricks” such as piloting the time increment in a smart way) either to reduce the CPU time needed in simulations of slender fluid domains (e.g. shock-tube like applications), or to enhance accuracy at a given computational cost.

Some corrections and ameliorations remain to be done:

- Full second order in space (option RECO 1) should be implemented for 1D VFCCs in order to



(a) Fluid pressure



(b) Fluid velocity

Figure 25: Further results of test SHOC65.

obtain maximum accuracy. Note that, since this option is global in EPX, when 1D VFCCs are present in a model, also the 2D or 3D VFCCs possibly present in the same model will be computed at first order in space.

- The printout of results for 1D VFCCs should be implemented.
- The interfacing of junction elements with the Cast3m mesh generator should be checked. It should be possible to visualize such elements in EPX without having to re-define them as element groups.

## References

- [1] EUROPLEXUS User's Manual, on-line version: <http://europlexus.jrc.ec.europa.eu>.
- [2] Cast3m Software: <http://www-cast3m.cea.fr/>.

# Appendix A — Input files

All the input files used in the previous Sections are listed below.

## 1d3d01.dgibi

```
opti echo 1;
opti dime 3 elem cub8;
opti trac psc ftra '1d3d01_mesh.ps';
opti sauv form '1d3d01.msh';
*
lhp = 16.0;
llp = lhp;
nhp = 1600;
nlp = nhp;
h = lhp / nhp;
tol = h / 10;
*
p0 = 0 0 0;
p1 = 0 h 0;
p2 = 0 h h;
p3 = 0 0 h;
bashp = manu qua4 p0 p1 p2 p3;
hp = bashp volu tran nhp (lhp 0 0);
baslp = bashp plus (lhp 0 0);
lp = baslp volu tran nlp (llp 0 0);
flui = hp et lp;
elim tol flui;
mesh = flui;
*
tass mesh noop;
sauv form mesh;
trac cach qual mesh;
*
fin;
```

## 1d3d01.epx

```
1D3D01
ECHO
!CONV win
CAST mesh
TRID EULE
GEOM CUVF flui TERM
COMP COUL ROUG LECT hp TERM
TURQ LECT lp TERM
MATE GAZP RO 14.4 GAMM 1.4 CV 720 PINI 12.E5 PREF 1.E5
LECT hp TERM
GAZP RO 1.20 GAMM 1.4 CV 720 PINI 1.0E5 PREF 1.E5
LECT lp TERM
ECRI TFRE 1.E-3
NOPO NOEL
FICH ALIC TFRE 1.E-3
OPTI NOTE CSTA 0.75
LOG 1
VFCC FC0N 6 ! hllc solver
ORDR 2 ! order in space
OTPS 2 ! order in time
RECO 1
CALC TINI 0 TEND 20.E-3
FIN
```

## 1d3d01p.epx

```
Post-treatment (space curves from alice file)
ECHO
RESU ALIC '1d3d01.ali' GARD PSCR
COMP NGRO 1 'xaxo' LECT flui TERM
COND LINE X1 0 Y1 0 Z1 0 X2 32 Y2 0 Z2 0 TOL 0.0001

SORT GRAP
PERF '1d3d01.pun'
AXTE 1.0 'Time [s]'
SCOU 61 'p_21' NSTO 21 SAXE 1.0 'curr_abcissa' LECT xaxo TERM
ECRO COMP 1
SCOU 62 'ro_21' NSTO 21 SAXE 1.0 'curr_abcissa' LECT xaxo TERM
ECRO COMP 2
SCOU 65 'vx_21' NSTO 21 SAXE 1.0 'curr_abcissa' LECT xaxo TERM
VCVI COMP 1
DCOU 71 'p_ana' SHTU GAMM 1.4 ROM 14.4 ROP 1.2 EINT 2.08333E5
LENM 16 LENP 16
TIME 20.E-3 NRAR 30 VARI 1
DCOU 72 'r_ana' SHTU GAMM 1.4 ROM 14.4 ROP 1.2 EINT 2.08333E5
LENM 16 LENP 16
TIME 20.E-3 NRAR 30 VARI 2
DCOU 75 'v_ana' SHTU GAMM 1.4 ROM 14.4 ROP 1.2 EINT 2.08333E5
LENM 16 LENP 16
TIME 20.E-3 NRAR 30 VARI 5

TRAC 61 71 AXES 1.0 'PRESS. [PA]'
COLO NOIR ROUG
LIST 61 AXES 1.0 'PRESS. [PA]'
TRAC 62 72 AXES 1.0 'DENS. [KG/M3]'
COLO NOIR ROUG
LIST 62 AXES 1.0 'DENS. [KG/M3]'
TRAC 65 75 AXES 1.0 'VELOC. [M/S]'
COLO NOIR ROUG
LIST 65 AXES 1.0 'VELOC. [M/S]'
FIN
```

## 1d3d02.dgibi

```
opti echo 1;
opti dime 2 elem qua4;
opti trac psc ftra '1d3d02_mesh.ps';
opti sauv form '1d3d02.msh';
*
lhp = 16.0;
llp = lhp;
nhp = 1600;
nlp = nhp;
h = lhp / nhp;
tol = h / 10;
*
p0 = 0 0;
p1 = 0 h;
bashp = manu seg2 p0 p1;
hp = bashp tran nhp (lhp 0);
baslp = bashp plus (lhp 0);
lp = baslp tran nlp (llp 0);
flui = hp et lp;
elim tol flui;
mesh = flui;
*
tass mesh noop;
sauv form mesh;
trac cach qual mesh;
*
fin;
```

## 1d3d02.epx

```
1D3D02
ECHO
!CONV win
CAST mesh
DPLA EULE
GEOM Q4VF flui TERM
COMP COUL ROUG LECT hp TERM
TURQ LECT lp TERM
MATE GAZP RO 14.4 GAMM 1.4 CV 720 PINI 12.E5 PREF 1.E5
LECT hp TERM
GAZP RO 1.20 GAMM 1.4 CV 720 PINI 1.0E5 PREF 1.E5
LECT lp TERM
ECRI TFRE 1.E-3
NOPO NOEL
FICH ALIC TFRE 1.E-3
OPTI NOTE CSTA 0.75
LOG 1
VFCC FC0N 6 ! hllc solver
ORDR 2 ! order in space
OTPS 2 ! order in time
RECO 1
CALC TINI 0 TEND 20.E-3
FIN
```

## 1d3d02p.epx

```
Post-treatment (space curves from alice file)
ECHO
RESU ALIC '1d3d02.ali' GARD PSCR
COMP NGRO 1 'xaxo' LECT flui TERM
COND LINE X1 0 Y1 0 X2 32 Y2 0 TOL 0.0001

SORT GRAP
PERF '1d3d02.pun'
AXTE 1.0 'Time [s]'
SCOU 61 'p_21' NSTO 21 SAXE 1.0 'curr_abcissa' LECT xaxo TERM
ECRO COMP 1
SCOU 62 'ro_21' NSTO 21 SAXE 1.0 'curr_abcissa' LECT xaxo TERM
ECRO COMP 2
SCOU 65 'vx_21' NSTO 21 SAXE 1.0 'curr_abcissa' LECT xaxo TERM
VCVI COMP 1
DCOU 71 'p_ana' SHTU GAMM 1.4 ROM 14.4 ROP 1.2 EINT 2.08333E5
LENM 16 LENP 16
TIME 20.E-3 NRAR 30 VARI 1
DCOU 72 'r_ana' SHTU GAMM 1.4 ROM 14.4 ROP 1.2 EINT 2.08333E5
LENM 16 LENP 16
TIME 20.E-3 NRAR 30 VARI 2
DCOU 75 'v_ana' SHTU GAMM 1.4 ROM 14.4 ROP 1.2 EINT 2.08333E5
LENM 16 LENP 16
TIME 20.E-3 NRAR 30 VARI 5

TRAC 61 71 AXES 1.0 'PRESS. [PA]'
COLO NOIR ROUG
LIST 61 AXES 1.0 'PRESS. [PA]'
TRAC 62 72 AXES 1.0 'DENS. [KG/M3]'
COLO NOIR ROUG
LIST 62 AXES 1.0 'DENS. [KG/M3]'
TRAC 65 75 AXES 1.0 'VELOC. [M/S]'
COLO NOIR ROUG
LIST 65 AXES 1.0 'VELOC. [M/S]'
FIN
```

## 1d3d03.dgibi

```
opti echo 1;
opti dime 3 elem cub8;
opti trac psc ftra '1d3d03_mesh.ps';
opti sauv form '1d3d03.msh';
*
lhp = 16.0;
llp = lhp;
nhp = 1600;
nlp = nhp;
*
p0 = 0 0 0;
p1 = lhp 0 0;
p2 = (lhp + llp) 0 0;
hp = p0 d nhp p1;
lp = p1 d nlp p2;
flui = hp et lp;
mesh = flui;
*
tass mesh noop;
sauv form mesh;
trac cach qual mesh;
*
fin;
```

## 1d3d03.epx

```
1D3D03
ECHO
!CONV win
CAST mesh
TRID EULE
GEOM TUVF flui TERM
COMP DIAM DROI 0.01 LECT flui TERM
      COUL ROUG LECT hp TERM
      TURQ LECT lp TERM
MATE GAZP RO 14.4 GAMM 1.4 CV 720 PINI 12.E5 PREF 1.E5
      LECT hp TERM
      GAZP RO 1.20 GAMM 1.4 CV 720 PINI 1.0E5 PREF 1.E5
      LECT lp TERM
ECRI TFRE 1.E-3
      NOPO NOEL
      FICH ALIC TFRE 1.E-3
OPTI NOTE CSTA 0.75
      LOG 1
      VFCC FC0N 6 ! hllc solver
      ORDR 2 ! order in space
      OTPS 2 ! order in time
      RECO 1 ! Not accepted by CAL_VFCC_1D
CALC TINI 0 TEND 20.E-3
FIN
```

## 1d3d03p.epx

```
Post-treatment (space curves from alice file)
ECHO
RESU ALIC '1d3d03.ali' GARD PSCR
COMP NGRO 1 'xaxo' LECT flui TERM
      COND LINE X1 0 Y1 0 Z1 0 X2 32 Y2 0 Z2 0 TOL 0.0001
SORT GRAP
PERF '1d3d03.pun'
AXTE 1.0 'Time [s]'
SCOU 61 'p_21' NSTO 21 SAXE 1.0 'curr_abcissa' LECT xaxo TERM
      ECR0 COMP 1
SCOU 62 'ro_21' NSTO 21 SAXE 1.0 'curr_abcissa' LECT xaxo TERM
      ECR0 COMP 2
SCOU 65 'vx_21' NSTO 21 SAXE 1.0 'curr_abcissa' LECT xaxo TERM
      VCVI COMP 1
DCOU 71 'p_ana' SHTU GAMM 1.4 ROM 14.4 ROP 1.2 EINT 2.08333E5
      LENM 16 LENP 16
      TIME 20.E-3 NRAR 30 VARI 1
DCOU 72 'r_ana' SHTU GAMM 1.4 ROM 14.4 ROP 1.2 EINT 2.08333E5
      LENM 16 LENP 16
      TIME 20.E-3 NRAR 30 VARI 2
DCOU 75 'v_ana' SHTU GAMM 1.4 ROM 14.4 ROP 1.2 EINT 2.08333E5
      LENM 16 LENP 16
      TIME 20.E-3 NRAR 30 VARI 5
TRAC 61 71 AXES 1.0 'PRESS. [PA]'
COLO NOIR ROUG
LIST 61 AXES 1.0 'PRESS. [PA]'
TRAC 62 72 AXES 1.0 'DENS. [KG/M3]'
COLO NOIR ROUG
LIST 62 AXES 1.0 'DENS. [KG/M3]'
TRAC 65 75 AXES 1.0 'VELOC. [M/S]'
COLO NOIR ROUG
LIST 65 AXES 1.0 'VELOC. [M/S]'
FIN
```

## 1d3d14.dgibi

```
opti echo 1;
opti dime 3 elem cub8;
opti trac psc ftra '1d3d14_mesh.ps';
opti sauv form '1d3d14.msh';
*
lhp3d = 12.0;
```

```
lhp1d = 4.0;
llp1d = 6.0;
llp3d = 10.0;
nhp3d = 1200;
nhp1d = 400;
nlp1d = 600;
nlp3d = 1000;
h = lhp3d / nhp3d;
tol = h / 10;
*
p0 = 0 0 0;
p1 = 0 h 0;
p2 = 0 h h;
p3 = 0 0 h;
bashp3d = manu qua4 p0 p1 p2 p3;
hp3d = bashp3d volu tran nhp3d (lhp3d 0 0);
baslp3d = bashp3d plus ((lhp3d+lhp1d+llp1d) 0 0);
lp3d = baslp3d volu tran nlp3d (llp3d 0 0);
pid1 = p0 plus (lhp3d 0 0);
pid2 = pid1 plus (lhp1d 0 0);
pid3 = pid2 plus (llp1d 0 0);
hpid = pid1 d nhpid pid2;
lpid = pid2 d nlpid pid3;
hp = hp3d et hpid;
lp = lp3d et lpid;
flui = hp et lp;
*
* raccords 3d-1d
*
facehp = bashp3d plus (lhp3d 0 0);
pfacehp = chan poi1 facehp;
elim tol (pfacehp et hp3d);
facelp = baslp3d;
pfacelp = chan poi1 facelp;
rachp = manu supe (pid1 et facehp);
raclp = manu supe (pid3 et facelp);
*
mesh = flui et facehp et facelp et rachp et raclp;
*
tass mesh noop;
sauv form mesh;
trac cach qual mesh;
*
fin;
```

## 1d3d14.epx

```
1D3D14
ECHO
!CONV win
CAST mesh
TRID EULE
DIME JONC 10 TERM ! Total n. of nodes in a TUBM juncton
GEOM CUVF hp3d lp3d TUVF hp1d lp1d CL3D facehp facelp
      TUBM rachp raclp TERM
COMP DIAM DROI 0.011283792 LECT hp1d lp1d TERM
      RACC TUBM LECT rachp TERM
      NTUB LECT pid1 TERM DTUB 0.011283792
      FACE LECT facehp TERM COEF 1.0
      RACC TUBM LECT raclp TERM
      NTUB LECT pid3 TERM DTUB 0.011283792
      FACE LECT facelp TERM COEF 1.0
! Attention: the TUBM elements (rachp and raclp) are NOT included
! in the "mesh" object (although they ARE indeed passed in from Cast3m).
! For this reason we must add them explicitly in the GROU directive below
! if we want to have them in the extracted element groups.
      GROU 2 'nrachp' LECT mesh rachp raclp TERM
      COND XB GT 11.99 COND XB LT 12.01
      'nraclp' LECT mesh rachp raclp TERM
      COND XB GT 21.99 COND XB LT 22.01
      COUL ROUG LECT hp TERM
      TURQ LECT lp TERM
      VERT LECT rachp raclp TERM
MATE GAZP RO 14.4 GAMM 1.4 CV 720 PINI 12.E5 PREF 1.E5
      LECT hp TERM
      GAZP RO 1.20 GAMM 1.4 CV 720 PINI 1.0E5 PREF 1.E5
      LECT lp TERM
! In order to obtain a printout at least of the 3D VFCCs I am obliged
! to use a different material for rachp and raclp, but with the
! same characteristics as the materials used for hp and lp, respectively
      GAZP RO 14.4 GAMM 1.4 CV 720 PINI 12.E5 PREF 1.E5
      LECT rachp TERM
      GAZP RO 1.20 GAMM 1.4 CV 720 PINI 1.0E5 PREF 1.E5
      LECT raclp TERM
ECRI ECR0 VFCC TFRE 1.E-3
! NOPO NOEL
      FICH ALIC TFRE 1.E-3
OPTI NOTE CSTA 0.75
      LOG 1
      VFCC FC0N 6 ! hllc solver
      ORDR 2 ! order in space
      OTPS 2 ! order in time
      RECO 1 ! Not accepted by CAL_VFCC_1D
CALC TINI 0 TEND 20.E-3
FIN
```

## 1d3d14p.epx

```

Post-treatment (space curves from alice file)
ECHO
OPTI PRIN
RESU ALIC '1d3d14.ali' GARD PSCR
COMP NGRO 1 'xaxo' LECT flui TERM
COND LINE X1 0 Y1 0 Z1 0 X2 32 Y2 0 Z2 0 TOL 0.0001

SORT GRAP
PERF '1d3d14.pun'
AXTE 1.0 'Time [s]'
SCOU 61 'p_21' NSTO 21 SAXE 1.0 'curr_abcissa' LECT xaxo TERM
ECRO COMP 1
SCOU 62 'ro_21' NSTO 21 SAXE 1.0 'curr_abcissa' LECT xaxo TERM
ECRO COMP 2
SCOU 65 'vx_21' NSTO 21 SAXE 1.0 'curr_abcissa' LECT xaxo TERM
VCVI COMP 1
DCOU 71 'p_ana' SHTU GAMM 1.4 ROM 14.4 ROP 1.2 EINT 2.08333E5
LENM 16 LENP 16
TIME 20.E-3 NRAR 30 VARI 1
DCOU 72 'r_ana' SHTU GAMM 1.4 ROM 14.4 ROP 1.2 EINT 2.08333E5
LENM 16 LENP 16
TIME 20.E-3 NRAR 30 VARI 2
DCOU 75 'v_ana' SHTU GAMM 1.4 ROM 14.4 ROP 1.2 EINT 2.08333E5
LENM 16 LENP 16
TIME 20.E-3 NRAR 30 VARI 5
TRAC 61 71 AXES 1.0 'PRESS. [PA]'
COLO NOIR ROUG
LIST 61 AXES 1.0 'PRESS. [PA]'
TRAC 62 72 AXES 1.0 'DENS. [KG/M3]'
COLO NOIR ROUG
LIST 62 AXES 1.0 'DENS. [KG/M3]'
TRAC 65 75 AXES 1.0 'VELOC. [M/S]'
COLO NOIR ROUG
LIST 65 AXES 1.0 'VELOC. [M/S]'
FIN

```

## 1d3d15.dgibi

```

opti echo 1;
opti dime 3 elem cub8;
opti trac psc ftra '1d3d15_mesh.ps';
opti sauv form '1d3d15.msh';
*
lhp3d = 12.0;
lhp1d = 4.0;
llp1d = 6.0;
llp3d = 10.0;
nhp3d = 1200;
nhp1d = 400;
nlp1d = 600;
nlp3d = 1000;
h = lhp3d / nhp3d;
tol = h / 10;
*
p0 = 0 0 0;
p1 = 0 h 0;
p2 = 0 h h;
p3 = 0 0 h;
bashp3d = manu qua4 p0 p1 p2 p3;
hp3d = bashp3d volu tran nhp3d (lhp3d 0 0);
baslp3d = bashp3d plus ((lhp3d+lhp1d+llp1d) 0 0);
lp3d = baslp3d volu tran nlp3d (llp3d 0 0);
pid1 = p0 plus (lhp3d 0 0);
pid2 = pid1 plus (lhp1d 0 0);
pid3 = pid2 plus (llp1d 0 0);
hpid = pid1 d nhpid pid2;
lpid = pid2 d nlpid pid3;
hp = hp3d et hpid;
lp = lp3d et lpid;
flui = hp et lp;
*
* raccords 3d-1d
*
facehp = bashp3d plus (lhp3d 0 0);
pfacehp = chan poi1 facehp;
elim tol (pfacehp et hp3d);
facelp = baslp3d;
pfacelp = chan poi1 facelp;
rachp = manu supe (pid1 et facehp);
raclp = manu supe (pid3 et facelp);
*
mesh = flui et facehp et facelp et rachp et raclp;
*
tass mesh noop;
sauv form mesh;
trac cach qual mesh;
*
fin;

```

## 1d3d15.epx

```

1D3D15
ECHO
!CONV win
CAST mesh
TRID EULE
DIME JONC 10 TERM ! Total n. of nodes in a TUBM juncton
GEOM CUVF hp3d lp3d TUVF hpid lpid CL3D facehp facelp
TUBM rachp raclp TERM
COMP DIAM DROI 0.01 LECT hpid lpid TERM

```

```

RACC TUBM LECT rachp TERM
NTUB LECT pid1 TERM DTUB 0.01
FACE LECT facehp TERM COEF 0.78539816
RACC TUBM LECT raclp TERM
NTUB LECT pid3 TERM DTUB 0.01
FACE LECT facelp TERM COEF 0.78539816
! Attention: the TUBM elements (rachp and raclp) are NOT included
! in the "mesh" object (although they ARE indeed passed in from Cast3m).
! For this reason we must add them explicitly in the GROU directive below
! if we want to have them in the extracted element groups.
GROU 2 'nrachp' LECT mesh rachp raclp TERM
COND XB GT 11.99 COND XB LT 12.01
'nraclp' LECT mesh rachp raclp TERM
COND XB GT 21.99 COND XB LT 22.01
COUL ROUG LECT hp TERM
TURQ LECT lp TERM
VERT LECT rachp raclp TERM
MATE GAZP RO 14.4 GAMM 1.4 CV 720 PINI 12.E5 PREF 1.E5
LECT hp TERM
GAZP RO 1.20 GAMM 1.4 CV 720 PINI 1.0E5 PREF 1.E5
LECT lp TERM
! In order to obtain a printout at least of the 3D VFCCs I am obliged
! to use a different material for rachp and raclp, but with the
! same characteristics as the materials used for hp and lp, respectively
GAZP RO 14.4 GAMM 1.4 CV 720 PINI 12.E5 PREF 1.E5
LECT rachp TERM
GAZP RO 1.20 GAMM 1.4 CV 720 PINI 1.0E5 PREF 1.E5
LECT raclp TERM
ECRI ECRO VFCC TFRE 1.E-3
! NOPO NOEL
FICH ALIC TFRE 1.E-3
OPTI NOTE CSTA 0.75
LOG 1
VFCC FCON 6 ! hllc solver
ORDR 2 ! order in space
OTPS 2 ! order in time
! RECO 1 ! Not accepted by CAL_VFCC_1D
CALC TINI 0 TEND 20.E-3
FIN

```

## 1d3d15p.epx

```

Post-treatment (space curves from alice file)
ECHO
OPTI PRIN
RESU ALIC '1d3d15.ali' GARD PSCR
COMP NGRO 1 'xaxo' LECT flui TERM
COND LINE X1 0 Y1 0 Z1 0 X2 32 Y2 0 Z2 0 TOL 0.0001

SORT GRAP
PERF '1d3d15.pun'
AXTE 1.0 'Time [s]'
SCOU 61 'p_21' NSTO 21 SAXE 1.0 'curr_abcissa' LECT xaxo TERM
ECRO COMP 1
SCOU 62 'ro_21' NSTO 21 SAXE 1.0 'curr_abcissa' LECT xaxo TERM
ECRO COMP 2
SCOU 65 'vx_21' NSTO 21 SAXE 1.0 'curr_abcissa' LECT xaxo TERM
VCVI COMP 1
DCOU 71 'p_ana' SHTU GAMM 1.4 ROM 14.4 ROP 1.2 EINT 2.08333E5
LENM 16 LENP 16
TIME 20.E-3 NRAR 30 VARI 1
DCOU 72 'r_ana' SHTU GAMM 1.4 ROM 14.4 ROP 1.2 EINT 2.08333E5
LENM 16 LENP 16
TIME 20.E-3 NRAR 30 VARI 2
DCOU 75 'v_ana' SHTU GAMM 1.4 ROM 14.4 ROP 1.2 EINT 2.08333E5
LENM 16 LENP 16
TIME 20.E-3 NRAR 30 VARI 5
TRAC 61 71 AXES 1.0 'PRESS. [PA]'
COLO NOIR ROUG
LIST 61 AXES 1.0 'PRESS. [PA]'
TRAC 62 72 AXES 1.0 'DENS. [KG/M3]'
COLO NOIR ROUG
LIST 62 AXES 1.0 'DENS. [KG/M3]'
TRAC 65 75 AXES 1.0 'VELOC. [M/S]'
COLO NOIR ROUG
LIST 65 AXES 1.0 'VELOC. [M/S]'
FIN

```

## 1d3d16.dgibi

```

opti echo 1;
opti dime 3 elem cub8;
opti trac psc ftra '1d3d16_mesh.ps';
opti sauv form '1d3d16.msh';
*
lhp3d = 12.0;
lhp1d = 4.0;
llp1d = 6.0;
llp3d = 10.0;
nhp3d = 1200;
nhp1d = 400;
nlp1d = 600;
nlp3d = 1000;
h = lhp3d / nhp3d;
tol = h / 10;
*
p0 = 0 0 0;
p1 = 0 h 0;
p2 = 0 h h;
p3 = 0 0 h;
bashp3d = manu qua4 p0 p1 p2 p3;
hp3d = bashp3d volu tran nhp3d (lhp3d 0 0);
baslp3d = bashp3d plus ((lhp3d+lhp1d+llp1d) 0 0);

```

```

lp3d = baslp3d volu tran nlp3d (llp3d 0 0);
pid1 = p0 plus (lhp3d 0 0);
pid2 = pid1 plus (lhp1d 0 0);
pid3 = pid2 plus (llp1d 0 0);
hpid = pid1 d nhpid pid2;
lp1d = pid2 d nlp1d pid3;
hp = hp3d et hpid;
lp = lp3d et lp1d;
flui = hp et lp;
*
* raccords 3d-1d
*
facehp = bashp3d plus (lhp3d 0 0);
pfacehp = chan poi1 facehp;
elim tol (pfacehp et hp3d);
facelp = baslp3d;
pfacelp = chan poi1 facelp;
rachp = manu supe (pid1 et facehp);
rac1p = manu supe (pid3 et facelp);
*
mesh = flui et facehp et facelp et rachp et rac1p;
*
tass mesh noop;
sauv form mesh;
trac cach qual mesh;
*
fin;

```

## 1d3d16.epx

```

1D3D16
ECHO
!CONV win
CAST mesh
TRID EULE
DIME JONC 10 TERM ! Total n. of nodes in a TUBM juncton
GEOM CUVF hp3d lp3d TUBE hpid lp1d CL3D facehp facelp
      TUBM rachp rac1p TERM
COMP DIAM DROI 0.011283792 LECT hpid lp1d TERM
      RACC TUBM LECT rachp TERM
      NTUB LECT pid1 TERM DTUB 0.011283792
      FACE LECT facehp TERM COEF 1.0
      RACC TUBM LECT rac1p TERM
      NTUB LECT pid3 TERM DTUB 0.011283792
      FACE LECT facelp TERM COEF 1.0
! Attention: the TUBM elements (rachp and rac1p) are NOT included
! in the "mesh" object (although they ARE indeed passed in from Cast3m).
! For this reason we must add them explicitly in the GROU directive below
! if we want to have them in the extracted element groups.
      GROU 2 'nrachp' LECT mesh rachp rac1p TERM
              COND XB GT 11.99 COND XB LT 12.01
              'nrac1p' LECT mesh rachp rac1p TERM
                      COND XB GT 21.99 COND XB LT 22.01
      COUL ROUG LECT hp TERM
      TURQ LECT lp TERM
              VERT LECT rachp rac1p TERM
MATE GAZP RO 14.4 GAMM 1.4 CV 720 PINI 12.E5 PREF 1.E5
      LECT hp TERM
      GAZP RO 1.20 GAMM 1.4 CV 720 PINI 1.0E5 PREF 1.E5
      LECT lp TERM
! In order to obtain a printout at least of the 3D VFCCs I am obliged
! to use a different material for rachp and rac1p, but with the
! same characteristics as the materials used for hp and lp, respectively
      GAZP RO 14.4 GAMM 1.4 CV 720 PINI 12.E5 PREF 1.E5
      LECT rachp TERM
      GAZP RO 1.20 GAMM 1.4 CV 720 PINI 1.0E5 PREF 1.E5
      LECT rac1p TERM
LIAI TUBM LECT rachp TERM
      TUBM LECT rac1p TERM
ECRI ECRO VFCC TFRE 1.E-3
!
      NOPO NOEL
      FICH ALIC TFRE 1.E-3
OPTI NOTE CSTA 0.75
      LOG 1
      VFCC FCON 6 ! hllc solver
              ORDR 2 ! order in space
              OTPS 2 ! order in time
!
      RECO 1 ! Not accepted by CAL_VFCC_1D
CALC TINI 0 TEND 20.E-3
FIN

```

## 1d3d16p.epx

```

Post-treatment (space curves from alice file)
ECHO
OPTI PRIN
RESU ALIC '1d3d16.ali' GARD PSCR
COMP NGRO 1 'xaxo' LECT flui TERM
      COND LINE X1 0 Y1 0 Z1 0 X2 32 Y2 0 Z2 0 TOL 0.0001
SORT GRAP
PERF '1d3d16.pun'
AXTE 1.0 'Time [s]'
SCOU 61 'p_21' NSTO 21 SAXE 1.0 'curr_abcissa' LECT xaxo TERM
      ECRO COMP 1
SCOU 62 'ro_21' NSTO 21 SAXE 1.0 'curr_abcissa' LECT xaxo TERM
      ECRO COMP 2
!SCOU 65 'vx_21' NSTO 21 SAXE 1.0 'curr_abcissa' LECT xaxo TERM
!
      VCVI COMP 1
DCOU 71 'p_ana' SHTU GAMM 1.4 ROM 14.4 ROP 1.2 EINT 2.08333E5

```

```

      LENM 16 LENP 16
      TIME 20.E-3 NRAR 30 VARI 1
DCOU 72 'r_ana' SHTU GAMM 1.4 ROM 14.4 ROP 1.2 EINT 2.08333E5
      LENM 16 LENP 16
      TIME 20.E-3 NRAR 30 VARI 2
DCOU 75 'v_ana' SHTU GAMM 1.4 ROM 14.4 ROP 1.2 EINT 2.08333E5
      LENM 16 LENP 16
      TIME 20.E-3 NRAR 30 VARI 5
TRAC 61 71 AXES 1.0 'PRESS. [PA]'
      COLO NOIR ROUG
      LIST 61 AXES 1.0 'PRESS. [PA]'
      TRAC 62 72 AXES 1.0 'DENS. [KG/M3]'
      COLO NOIR ROUG
      LIST 62 AXES 1.0 'DENS. [KG/M3]'
      !TRAC 65 75 AXES 1.0 'VELOC. [M/S]'
      !COLO NOIR ROUG
      !LIST 65 AXES 1.0 'VELOC. [M/S]'
FIN

```

## 1d3d17.dgibi

```

opti echo 1;
opti dime 3 elem cub8;
opti trac psc ftra '1d3d17_mesh.ps';
opti sauv form '1d3d17.msh';
*
lhp3d = 12.0;
lhp1d = 4.0;
llp1d = 6.0;
llp3d = 10.0;
nhp3d = 1200;
nhp1d = 400;
nlp1d = 600;
nlp3d = 1000;
h = lhp3d / nhp3d;
tol = h / 10;
*
p0 = 0 0 0;
p1 = 0 h 0;
p2 = 0 h h;
p3 = 0 0 h;
bashp3d = manu qua4 p0 p1 p2 p3;
hp3d = bashp3d volu tran nhp3d (lhp3d 0 0);
baslp3d = bashp3d plus ((lhp3d+lhp1d+llp1d) 0 0);
lp3d = baslp3d volu tran nlp3d (llp3d 0 0);
pid1 = p0 plus (lhp3d 0 0);
pid2 = pid1 plus (lhp1d 0 0);
pid3 = pid2 plus (llp1d 0 0);
hpid = pid1 d nhpid pid2;
lp1d = pid2 d nlp1d pid3;
hp = hp3d et hpid;
lp = lp3d et lp1d;
flui = hp et lp;
*
* raccords 3d-1d
*
facehp = bashp3d plus (lhp3d 0 0);
pfacehp = chan poi1 facehp;
elim tol (pfacehp et hp3d);
facelp = baslp3d;
pfacelp = chan poi1 facelp;
rachp = manu supe (pid1 et facehp);
rac1p = manu supe (pid3 et facelp);
*
mesh = flui et facehp et facelp et rachp et rac1p;
*
tass mesh noop;
sauv form mesh;
trac cach qual mesh;
*
fin;

```

## 1d3d17.epx

```

1D3D17
ECHO
!CONV win
CAST mesh
TRID EULE
DIME JONC 10 TERM ! Total n. of nodes in a TUBM juncton
GEOM CUVF hp3d lp3d TUBE hpid lp1d CL3D facehp facelp
      TUBM rachp rac1p TERM
COMP DIAM DROI 0.011283792 LECT hpid lp1d TERM
      RACC TUBM LECT rachp TERM
      NTUB LECT pid1 TERM DTUB 0.011283792
      FACE LECT facehp TERM COEF 1.0
      RACC TUBM LECT rac1p TERM
      NTUB LECT pid3 TERM DTUB 0.011283792
      FACE LECT facelp TERM COEF 1.0
! Attention: the TUBM elements (rachp and rac1p) are NOT included
! in the "mesh" object (although they ARE indeed passed in from Cast3m).
! For this reason we must add them explicitly in the GROU directive below
! if we want to have them in the extracted element groups.
      GROU 2 'nrachp' LECT mesh rachp rac1p TERM
              COND XB GT 11.99 COND XB LT 12.01
              'nrac1p' LECT mesh rachp rac1p TERM
                      COND XB GT 21.99 COND XB LT 22.01
      COUL ROUG LECT hp TERM
      TURQ LECT lp TERM
              VERT LECT rachp rac1p TERM
MATE GAZP RO 14.4 GAMM 1.4 CV 720 PINI 12.E5 PREF 1.E5
      LECT hp TERM

```

```

GAZP RD 1.20 GAMM 1.4 CV 720 PINI 1.0E5 PREF 1.E5
LECT lp TERM
! In order to obtain a printout at least of the 3D VFCCs I am obliged
! to use a different material for rachp and raclp, but with the
! same characteristics as the materials used for hp and lp, respectively
GAZP RD 14.4 GAMM 1.4 CV 720 PINI 12.E5 PREF 1.E5
LECT rachp TERM
GAZP RD 1.20 GAMM 1.4 CV 720 PINI 1.0E5 PREF 1.E5
LECT raclp TERM
LIAI TUBM LECT rachp TERM
TUBM LECT raclp TERM
ECRI ECRO VFCC TFRE 1.E-3
! NOPO NOEL
FICH ALIC TFRE 1.E-3
OPTI NOTE CSTA 0.75
LOG 1
VFCC FCON 6 ! hllc solver
ORDR 2 ! order in space
OTPS 2 ! order in time
RECO 1 ! true second order in space
CALC TINI 0 TEND 20.E-3
FIN

```

## 1d3d17p.epx

```

Post-treatment (space curves from alice file)
ECHO
OPTI PRIN
RESU ALIC '1d3d17.ali' GARD PSCR
COMP NGRO 1 'xaxo' LECT flui TERM
COND LINE X1 0 Y1 0 Z1 0 X2 32 Y2 0 Z2 0 TOL 0.0001

SORT GRAP
PERF '1d3d17.pun'
AXTE 1.0 'Time [s]'
SCOU 61 'p_21' NSTO 21 SAXE 1.0 'curr_abcissa' LECT xaxo TERM
ECRO COMP 1
SCOU 62 'ro_21' NSTO 21 SAXE 1.0 'curr_abcissa' LECT xaxo TERM
ECRO COMP 2
!SCOU 65 'vx_21' NSTO 21 SAXE 1.0 'curr_abcissa' LECT xaxo TERM
! VCVI COMP 1
DCOU 71 'p_ana' SHTU GAMM 1.4 ROM 14.4 ROP 1.2 EINT 2.08333E5
LENM 16 LENP 16
TIME 20.E-3 NRAR 30 VARI 1
DCOU 72 'r_ana' SHTU GAMM 1.4 ROM 14.4 ROP 1.2 EINT 2.08333E5
LENM 16 LENP 16
TIME 20.E-3 NRAR 30 VARI 2
DCOU 75 'v_ana' SHTU GAMM 1.4 ROM 14.4 ROP 1.2 EINT 2.08333E5
LENM 16 LENP 16
TIME 20.E-3 NRAR 30 VARI 5
TRAC 61 71 AXES 1.0 'PRESS. [PA]'
COLO NOIR ROUG
LIST 61 AXES 1.0 'PRESS. [PA]'
TRAC 62 72 AXES 1.0 'DENS. [KG/M3]'
COLO NOIR ROUG
LIST 62 AXES 1.0 'DENS. [KG/M3]'
!TRAC 65 75 AXES 1.0 'VELOC. [M/S]'
!COLO NOIR ROUG
!LIST 65 AXES 1.0 'VELOC. [M/S]'
FIN

```

## shoc50.dgibi

```

opti echo 1;
opti dime 3 elem cub8;
opti trac psc ftra 'shoc50_mesh.ps';
opti sauv form 'shoc50.msh';
*
dpla = 0.3;
rpla = 0.5 * dpla;
dext = 0.625;
rext = 0.5 * dext;
dtank = 1.6;
rtank = 0.5 * dtank;
ltan0 = -0.7;
ltan1 = 0.9;
ltanopen = 1.4;
ltan2 = 1.5;
*
ltubhp = 2;
ltublp = 16.20;
ltube = ltubhp + ltublp;
*
nrpla = 3;
tol = 0.01 * dpla / nrpla;
*
p0s = 0 0 0;
p1s = 0 rpla 0;
p2s = 0 rpla rpla;
p3s = 0 0 rpla;
*
c1s = p0s d nrpla p1s;
c2s = p1s d nrpla p2s;
c3s = p2s d nrpla p3s;
c4s = p3s d nrpla p0s;
plat = dall c1s c2s c3s c4s plan;
*
p0 = p0s plus p0s;
p1 = p1s plus p0s;
p2 = p2s plus p0s;

```

```

p3 = p3s plus p0s;
*
c1 = p0 d nrpla p1;
c2 = p1 d nrpla p2;
c3 = p2 d nrpla p3;
c4 = p3 d nrpla p0;
sflu1 = dall c1 c2 c3 c4 plan;
*
paxis = 1 0 0;
p4 = 0 rext 0;
p5 = p4 tour 45 p0 paxis;
p6 = 0 0 rext;
*
c5 = p1 d nrpla p4;
c6 = p4 c nrpla p0 p5;
c7 = p5 d nrpla p2;
c8 = p2 d nrpla p1;
sflu2 = dall c5 c6 c7 c8 plan;
*
c9 = p2 d nrpla p5;
c10 = p5 c nrpla p0 p6;
c11 = p6 d nrpla p3;
c12 = p3 d nrpla p2;
sflu3 = dall c9 c10 c11 c12 plan;
*
p7 = 0 rtank 0;
p8 = p7 tour 45 p0 paxis;
p9 = 0 0 rtank;
*
din = (rext - rpla) / nrpla;
dfi = din * dtank / dext;
c13 = p4 d p7 dini din dfi dfi;
sflu4 = c13 rota 3 45 p0 paxis;
*
sflu5 = sflu4 tour 45 p0 paxis;
*
sflu0 = sflu4 et sflu5;
sflu = sflu1 et sflu2 et sflu3 et sflu0;
elim tol sflu;
*
d1 = 0.05;
d2 = d1 * 2;
d3 = d2 * 2;
vtan0 = (sflu0 plus ((ltan0 0 0)) volu dini d2 dfin d1
tran ((0-ltan0) 0 0));
vtan1 = sflu volu dini d1 dfin d2 tran (ltan1 0 0);
vtanopen = (sflu plus (ltan1 0 0)) volu dini d2 dfin d3
tran (ltanopen 0 0);
vtan2 = (sflu plus ((ltan1+ltanopen) 0 0)) volu dini d3 dfin d3
tran (ltan2 0 0);
tank = vtan0 et vtan1 et vtanopen et vtan2;
elim tol tank;
*
pa1 = ltan1 rtank 0;
pa2 = (ltan1 + ltanopen) rtank 0;
cab = d pa1 pa2 dini d2 dfin d3;
abso = cab rota (2*nrpla) 90 p0 paxis;
elim tol (abso et tank);
*
tubehp = (sflu1 plus ((0 - ltube) 0 0)) volu dini d1 dfin d1
tran (ltubhp 0 0);
tubelp = (sflu1 plus ((0 - ltublp) 0 0)) volu dini d1 dfin d1
tran (ltublp 0 0);
tube = tubehp et tubelp;
*
flui = tube et tank;
elim tol flui;
trac cach tank;
trac cach flui;
*
list (nbel flui);
list (nbno flui);
list (nbel tube);
list (nbel tank);
list (nbel plat);
list (nbno plat);
*
mesh = flui et plat et abso;
tass mesh noop;
sauv form mesh;
*
fin;

```

## shoc50.epx

```

SHOC50
ECHO
!CONV win
CAST mesh
TRID ALE EROS 1.0
DIME NALE 1 NBLE 1 TERM
GEOM CUVF flui Q4GS plat CL3D abso TERM
COMP GROU 3 'endtube' LECT tube TERM COND XB GT -0.6
'trigger' LECT tube TERM COND NEAR POIN -1.01 0 0
'frefine' LECT flui TERM COND XB GT -1.2
COND XB LT 1.0
EPAI 0.8E-3 LECT plat TERM
COUL ROUG LECT tubehp TERM
TURQ LECT tubelp tank DIFF trigger TERM
VERT LECT plat TERM
ROSE LECT trigger TERM

```



```

JAUN LECT abso          TERM
GRIL LAGR LECT plat TERM
EULE LECT flui TERM
MATE GAZP RO 59.459 GAMM 1.4 CV 719.286 PINI 50.E5 PREF 1.E5
      LECT tubehp TERM
      GAZP RO 1.189 GAMM 1.4 CV 719.286 PINI 1.E5 PREF 1.E5
      LECT tubelp tank TERM
      VPJC RO 2700.0 YOUN 70.0E9 NU 0.3 ELAS 97.6E6 MXIT 200
      QR1 13.2E6 CR1 1458.0 QR2 10.6E6 CR2 185.9
      PDOT 5.E-4 C 1.4E-2 TQ 0.9 CP 910.0
      TM 893.0 M 1.0 DC 1.0 WC 44.6E6
      LECT plat TERM
      CLVF ABSO RO 1.20
      LECT abso TERM
LINK COUP
      BLOQ 123456 LECT c2s c3s TERM
!
! cont spla not yet available in adaptivity for boundary-hanging nodes!
! use explicit blockages instead
! CONT SPLA NX 0 NY 0 NZ 1 LECT c1s DIFF p1s TERM
! CONT SPLA NX 0 NY 1 NZ 0 LECT c4s DIFF p3s TERM
      BLOQ 345 LECT c1s DIFF p0s p1s TERM
      BLOQ 246 LECT c4s DIFF p0s p3s TERM
      BLOQ 23456 LECT p0s TERM
LINK DECO FLSW STRU LECT plat TERM
      FLUI LECT flui TERM
      R      0.0436 ! 0.87*H_FLUID = 0.87*0.05 = 0.0435
      HGRI 0.06 ! Slightly bigger than structural element 0.05
!
! DGR1
! FACE
! BFLU 2 ! block if at least one node is in influence domain
! FSCP 1 ! couple in all directions
ECRI DEPL VITE TFRE 1.E-3
      POIN LECT p0s TERM NOEL
      FICH SPLI ALIC FREQ 0 TFRE 0.DO
      TIME PRDG 0.DO PAS 1.D-3 30.D-3 PAS 0.1D-3 40.D-3
      PAS 1.D-3 80.D-3
!
OPTI NOTE CSTA 0.5
      LOG 1
      REZO GAMO 0.5
      VFCC FCON 6 ! hllc solver
      ORDR 2 ! order in space
      OTPS 2 ! order in time
!
! RECO 1 ! To compare with test SHOC61
CALC TINI 0 TEND 23.E-3
FIN

```

## shoc50b.epx

```

Post treatment (visualization from alice file)
ECHO
RESU SPLI ALIC 'shoc50.ali' GARD PSCR
OPTI PRIN
SORT VISU NSTO 24
PLAY
CAME 6 EYE -1.00291E-01 -3.09784E+00 -3.09784E+00
!
      Q      3.82683E-01 9.23880E-01 0.00000E+00 0.00000E+00
      VIEW 0.00000E+00 7.07107E-01 7.07107E-01
      RIGH 1.00000E+00 0.00000E+00 0.00000E+00
      UP    0.00000E+00 -7.07107E-01 7.07107E-01
      FOV   2.48819E+01
!NAVIGATION MODE: ROTATING CAMERA
!CENTER : -1.00291E-01 4.00000E-01 4.00000E-01
!RSPHERE: 1.23667E+00
!RADIUS : 4.94669E+00
!ASPECT : 1.00000E+00
!NEAR : 3.71002E+00
!FAR : 7.42004E+00
CAME 5 EYE 5.61480E-02 -5.86912E-01 -5.86912E-01
!
      Q      3.82683E-01 9.23880E-01 0.00000E+00 0.00000E+00
      VIEW 0.00000E+00 7.07107E-01 7.07107E-01
      RIGH 1.00000E+00 0.00000E+00 0.00000E+00
      UP    0.00000E+00 -7.07107E-01 7.07107E-01
      FOV   2.48819E+01
!NAVIGATION MODE: ROTATING CAMERA
!CENTER : 5.61480E-02 7.50000E-02 7.50000E-02
!RSPHERE: 1.20011E-01
!RADIUS : 9.36084E-01
!ASPECT : 1.00000E+00
!NEAR : 8.16074E-01
!FAR : 1.17611E+00
SCEN GEOM NAVI FREE
      ISO FILL FIEL ECRO 1 SUPP LECT flui TERM
      LIMA ON
      SLER CAM1 6 NFRA 1
      TRAC OFFS FICH BMP OBJE LECT flui TERM REND
      SCEN OBJE USLM LECT flui TERM
      GEOM NAVI FREE
      FACE SBAC
      LINE SSHA SFRE
      VECT SOCO FIEL VITE SUPP LECT plat TERM
      LENG 0.1
      LIMA ON
      SLER CAM1 5 NFRA 1
      TRAC OFFS FICH BMP OBJE LECT flui plat TERM NFAI REND
      ENDPLAY
      FIN

```

## shoc50p.epx

```

Post-treatment (space curves from alice file)
ECHO
RESU SPLI ALIC 'shoc50.ali' GARD PSCR
COMP NGRO 1 'xaxo' LECT flui TERM
      COND LINE X1 -18.2 Y1 0 Z1 0
      X2 3.8 Y2 0 Z2 0 TOL 0.0001
!
! SORT GRAP
! PERF 'shoc50p.pun'
! AXTE 1.0 'Time [s]'
! SCOU 51 'p_21' NSTO 21 SAXE 1.0 'curr_abcissa' LECT xaxo TERM
!      ECRO COMP 1
! SCOU 52 'ro_21' NSTO 21 SAXE 1.0 'curr_abcissa' LECT xaxo TERM
!      ECRO COMP 2
! SCOU 55 'vx_21' NSTO 21 SAXE 1.0 'curr_abcissa' LECT xaxo TERM
!      VCVI COMP 1
! SCOU 61 'p_24' NSTO 24 SAXE 1.0 'curr_abcissa' LECT xaxo TERM
!      ECRO COMP 1
! SCOU 62 'ro_24' NSTO 24 SAXE 1.0 'curr_abcissa' LECT xaxo TERM
!      ECRO COMP 2
! SCOU 65 'vx_24' NSTO 24 SAXE 1.0 'curr_abcissa' LECT xaxo TERM
!      VCVI COMP 1
! TRAC 51 61 AXES 1.0 'PRESS. [PA]' XGRD YGRD
! COLO ROUG NOIR
! LIST 51 61 AXES 1.0 'PRESS. [PA]'
! TRAC 52 62 AXES 1.0 'DENS. [KG/M3]' XGRD YGRD
! COLO ROUG NOIR
! LIST 52 62 AXES 1.0 'DENS. [KG/M3]'
! TRAC 55 65 AXES 1.0 'VELOC. [M/S]' YZER XGRD YGRD
! COLO ROUG NOIR
! LIST 55 65 AXES 1.0 'VELOC. [M/S]'
! FIN

```

## shoc50z.epx

```

Post treatment (visualization from alice file)
ECHO
RESU SPLI ALIC 'shoc50.ali' GARD PSCR
OPTI PRIN
SORT VISU NSTO 24
FIN

```

## shoc51.dgibi

```

opti echo 1;
opti dime 3 elem cub8;
opti trac psc ftra 'shoc51_mesh.ps';
opti sauv form 'shoc51.msh';
*
dpla = 0.3;
rpla = 0.5 * dpla;
dext = 0.625;
rext = 0.5 * dext;
dtank = 1.6;
rtank = 0.5 * dtank;
ltan0 = -0.7;
ltan1 = 0.9;
ltanopen = 1.4;
ltan2 = 1.5;
*
ltubhp = 2;
ltublp = 16.20;
ltube = ltubhp + ltublp;
*
nrpla = 3;
tol = 0.01 * dpla / nrpla;
*
p0s = 0 0 0;
p1s = 0 rpla 0;
p2s = 0 rpla rpla;
p3s = 0 0 rpla;
*
c1s = p0s d nrpla p1s;
c2s = p1s d nrpla p2s;
c3s = p2s d nrpla p3s;
c4s = p3s d nrpla p0s;
plat = dall cis c2s c3s c4s plan;
*
p0 = p0s plus p0s;
p1 = p1s plus p0s;
p2 = p2s plus p0s;
p3 = p3s plus p0s;
*
c1 = p0 d nrpla p1;
c2 = p1 d nrpla p2;
c3 = p2 d nrpla p3;
c4 = p3 d nrpla p0;
sflu1 = dall c1 c2 c3 c4 plan;
*
paxis = 1 0 0;
p4 = 0 rext 0;
p5 = p4 tour 45 p0 paxis;
p6 = 0 0 rext;
*
c5 = p1 d nrpla p4;
c6 = p4 c nrpla p0 p5;
c7 = p5 d nrpla p2;
c8 = p2 d nrpla p1;
sflu2 = dall c5 c6 c7 c8 plan;
*
c9 = p2 d nrpla p5;
c10 = p5 c nrpla p0 p6;

```

```

c11 = p6 d nrpla p3;
c12 = p3 d nrpla p2;
sflu3 = dall c9 c10 c11 c12 plan;
*
p7 = 0 rtank 0;
p8 = p7 tour 45 p0 paxis;
p9 = 0 0 rtank;
*
din = (rext - rpla) / nrpla;
dfi = din * dtank / dext;
c13 = p4 d p7 dini din dfi;
sflu4 = c13 rota 3 45 p0 paxis;
*
sflu5 = sflu4 tour 45 p0 paxis;
*
sflu0 = sflu4 et sflu5;
sflu = sflu1 et sflu2 et sflu3 et sflu0;
elim tol sflu;
*
d1 = 0.05;
d2 = d1 * 2;
d3 = d2 * 2;
vtan0 = (sflu0 plus (ltan0 0 0)) volu dini d2 dfi d1
      tran ((0-ltan0) 0 0);
vtan1 = sflu volu dini d1 dfi d2 tran (ltan1 0 0);
vtanopen = (sflu plus (ltan1 0 0)) volu dini d2 dfi d3
      tran (ltanopen 0 0);
vtan2 = (sflu plus ((ltan1+ltanopen) 0 0)) volu dini d3 dfi d3
      tran (ltan2 0 0);
tank = vtan0 et vtan1 et vtanopen et vtan2;
elim tol tank;
*
pa1 = ltan1 rtank 0;
pa2 = (ltan1 + ltanopen) rtank 0;
cab = d pa1 pa2 dini d2 dfi d3;
abso = cab rota (2*nrpla) 90 p0 paxis;
elim tol (abso et tank);
*
tubehp = (sflu1 plus ((0 - ltube) 0 0)) volu dini d1 dfi d1
      tran (ltubhp 0 0);
tubelp = (sflu1 plus ((0 - ltublp) 0 0)) volu dini d1 dfi d1
      tran (ltublp 0 0);
tube = tubehp et tubelp;
*
flui = tube et tank;
elim tol flui;
trac cach tank;
trac cach flui;
*
list (nbcl flui);
list (nbmo flui);
list (nbcl tube);
list (nbcl tank);
list (nbcl plat);
list (nbmo plat);
*
mesh = flui et plat et abso;
tass mesh noop;
sauv form mesh;
*
fin;

```

## shoc51.epx

```

SHOC51
ECHO
!CONV win
CAST mesh
TRID ALE EROS 1.0
DIME NALE 1 NBLE 1 TERM
GEOM CUVF flui Q4GS plat CL3D abso TERM
COMP GROU 3 'endtube' LECT tube TERM COND XB GT -0.6
          'trigger' LECT tube TERM COND NEAR POIN -1.01 0 0
          'frefine' LECT flui TERM COND XB GT -1.2
                  COND XB LT 1.0
          EPAI 0.8E-3 LECT plat TERM
          COUL ROUG LECT tubehp TERM
          TURQ LECT tubelp tank DIFF trigger TERM
          VERT LECT plat TERM
          ROSE LECT trigger TERM
          JAUN LECT abso TERM
GRIL LAGR LECT plat TERM
EULE LECT flui TERM
MATE GAZP RO 59.459 GAMM 1.4 CV 719.286 PINI 50.E5 PREF 1.E5
      LECT tubehp TERM
      GAZP RO 1.189 GAMM 1.4 CV 719.286 PINI 1.E5 PREF 1.E5
      LECT tubelp tank TERM
      VPJC RO 2700.0 YOUN 70.0E9 NU 0.3 ELAS 97.6E6 MXIT 200
      QR1 13.2E6 CR1 1458.0 QR2 10.6E6 CR2 185.9
      PDOT 5.E-4 C 1.4E-2 TQ 0.9 CP 910.0
      TM 893.0 M 1.0 DC 1.0 WC 44.6E6
      LECT plat TERM
      CLVF ABSO RO 1.20
      LECT abso TERM
LINK COUP
      BLOQ 123456 LECT c2s c3s TERM
!
! cont spla not yet available in adaptivity for boundary-hanging nodes!
! use explicit blockages instead
!   CONT SPLA NX 0 NY 0 NZ 1 LECT c1s DIFF p1s TERM
!   CONT SPLA NX 0 NY 1 NZ 0 LECT c4s DIFF p3s TERM

```

```

BLOQ 345 LECT c1s DIFF p0s p1s TERM
BLOQ 246 LECT c4s DIFF p0s p3s TERM
BLOQ 23456 LECT p0s TERM
LINK DECO FLSW STRU LECT plat TERM
      FLUI LECT flui TERM
      R 0.0436 ! 0.87*H_FLUID = 0.87*0.05 = 0.0435
      HGRI 0.06 ! Slightly bigger than structural element 0.05
      DGRI
      FACE
      BFLU 2 ! block if at least one node is in influence domain
      FSCP 1 ! couple in all directions
ECRI DEPL VITE TFRE 1.E-3
      POIN LECT p0s TERM NOEL
      FICH SPLI ALIC FREQ 0 TFRE 0.D0
      TIME PROG 0.D0 PAS 1.D-3 30.D-3 PAS 0.1D-3 40.D-3
      PAS 1.D-3 80.D-3
OPTI NOTE CSTA 0.5
      LOG 1
      REZO GAMO 0.5
      VFCC FCON 6 ! hllc solver
      ORDR 2 ! order in space
      OTPS 2 ! order in time
      RECO 1
CALC TINI 0 TEND 23.E-3
FIN

```

## shoc51b.epx

```

Post treatment (visualization from alice file)
ECHO
RESU SPLI ALIC 'shoc51.ali' GARD PSCR
OPTI PRIN
SORT VISU NSTO 24
PLAY
CAME 6 EYE -1.00291E-01 -3.09784E+00 -3.09784E+00
! Q 3.82683E-01 9.23880E-01 0.00000E+00 0.00000E+00
  VIEW 0.00000E+00 7.07107E-01 7.07107E-01
  RIGH 1.00000E+00 0.00000E+00 0.00000E+00
  UP 0.00000E+00 -7.07107E-01 7.07107E-01
  FOV 2.48819E+01
!NAVIGATION MODE: ROTATING CAMERA
!CENTER : -1.00291E-01 4.00000E-01 4.00000E-01
!RSPHERE: 1.23667E+00
!RADIUS : 4.94669E+00
!ASPECT : 1.00000E+00
!NEAR : 3.71002E+00
!FAR : 7.42004E+00
CAME 5 EYE 5.61480E-02 -5.86912E-01 -5.86912E-01
! Q 3.82683E-01 9.23880E-01 0.00000E+00 0.00000E+00
  VIEW 0.00000E+00 7.07107E-01 7.07107E-01
  RIGH 1.00000E+00 0.00000E+00 0.00000E+00
  UP 0.00000E+00 -7.07107E-01 7.07107E-01
  FOV 2.48819E+01
!NAVIGATION MODE: ROTATING CAMERA
!CENTER : 5.61480E-02 7.50000E-02 7.50000E-02
!RSPHERE: 1.20011E-01
!RADIUS : 9.36084E-01
!ASPECT : 1.00000E+00
!NEAR : 8.16074E-01
!FAR : 1.17611E+00
SCEN GEOM NAVI FREE
      ISO FILL FIEL ECR0 1 SUPP LECT flui TERM
      LIMA ON
      SLER CAM1 6 NFRA 1
      TRAC OFFS FICH BMP OBJE LECT flui TERM REND
      SCEN OBJE USLM LECT flui TERM
      GEOM NAVI FREE
      FACE SBAC
      LINE SSHA SFRE
      VECT SCCO FIEL VITE SUPP LECT plat TERM
      LENG 0.1
      LIMA ON
      SLER CAM1 5 NFRA 1
      TRAC OFFS FICH BMP OBJE LECT flui plat TERM NFAI REND
      ENDPLAY
      FIN

```

## shoc51p.epx

```

Post-treatment (space curves from alice file)
ECHO
RESU SPLI ALIC 'shoc51.ali' GARD PSCR
COMP NGRO 1 'xaxo' LECT flui TERM
      COND LINE X1 -18.2 Y1 0 Z1 0
      X2 3.8 Y2 0 Z2 0 TOL 0.0001
SORT GRAP
PERF 'shoc51p.pun'
AXTE 1.0 'Time [s]'
SCOU 51 'p_21' NSTO 21 SAXE 1.0 'curr_abcissa' LECT xaxo TERM
      ECR0 COMP 1
SCOU 52 'ro_21' NSTO 21 SAXE 1.0 'curr_abcissa' LECT xaxo TERM
      ECR0 COMP 2
SCOU 55 'vx_21' NSTO 21 SAXE 1.0 'curr_abcissa' LECT xaxo TERM
      VCVI COMP 1
SCOU 61 'p_24' NSTO 24 SAXE 1.0 'curr_abcissa' LECT xaxo TERM
      ECR0 COMP 1
SCOU 62 'ro_24' NSTO 24 SAXE 1.0 'curr_abcissa' LECT xaxo TERM
      ECR0 COMP 2
SCOU 65 'vx_24' NSTO 24 SAXE 1.0 'curr_abcissa' LECT xaxo TERM
      VCVI COMP 1
TRAC 51 61 AXES 1.0 'PRESS. [PA]' XGRD YGRD
COLO ROUG NOIR

```

```

LIST 51 61 AXES 1.0 'PRESS. [PA]'
TRAC 52 62 AXES 1.0 'DENS. [KG/M3]' XGRD YGRD
COLO ROUG NOIR
LIST 52 62 AXES 1.0 'DENS. [KG/M3]'
TRAC 55 65 AXES 1.0 'VELOC. [M/S]' YZER XGRD YGRD
COLO ROUG NOIR
LIST 55 65 AXES 1.0 'VELOC. [M/S]'
FIN

```

## shoc51z.epx

```

Post treatment (visualization from alice file)
ECHO
RESU SPLI ALIC 'shoc51.ali' GARD PSCR
OPTI PRIN
SORT VISU NSTO 24
FIN

```

## shoc52.dgibi

```

opti echo 1;
opti dime 3 elem cub8;
opti trac psc ftra 'shoc52_mesh.ps';
opti sauv form 'shoc52.msh';
*
dpla = 0.3;
rpla = 0.5 * dpla;
dext = 0.625;
rext = 0.5 * dext;
dtank = 1.6;
rtank = 0.5 * dtank;
ltan0 = -0.7;
ltan1 = 0.9;
ltanopen = 1.4;
ltan2 = 1.5;
*
ltubhp = 2;
ltublp = 16.20;
ltube = ltubhp + ltublp;
*
nrpla = 3;
tol = 0.01 * dpla / nrpla;
*
p0s = 0 0 0;
p1s = 0 rpla 0;
p2s = 0 rpla rpla;
p3s = 0 0 rpla;
*
c1s = p0s d nrpla p1s;
c2s = p1s d nrpla p2s;
c3s = p2s d nrpla p3s;
c4s = p3s d nrpla p0s;
plat = dall c1s c2s c3s c4s plan;
*
p0 = p0s plus p0s;
p1 = p1s plus p0s;
p2 = p2s plus p0s;
p3 = p3s plus p0s;
*
c1 = p0 d nrpla p1;
c2 = p1 d nrpla p2;
c3 = p2 d nrpla p3;
c4 = p3 d nrpla p0;
sflu1 = dall c1 c2 c3 c4 plan;
*
paxis = 1 0 0;
p4 = 0 rext 0;
p5 = p4 tour 45 p0 paxis;
p6 = 0 0 rext;
*
c5 = p1 d nrpla p4;
c6 = p4 c nrpla p0 p5;
c7 = p5 d nrpla p2;
c8 = p2 d nrpla p1;
sflu2 = dall c5 c6 c7 c8 plan;
*
c9 = p2 d nrpla p5;
c10 = p5 c nrpla p0 p6;
c11 = p6 d nrpla p3;
c12 = p3 d nrpla p2;
sflu3 = dall c9 c10 c11 c12 plan;
*
p7 = 0 rtank 0;
p8 = p7 tour 45 p0 paxis;
p9 = 0 0 rtank;
*
din = (rext - rpla) / nrpla;
dfi = din * dtank / dext;
c13 = p4 d p7 dini din dfi dfi;
sflu4 = c13 rota 3 45 p0 paxis;
*
sflu5 = sflu4 tour 45 p0 paxis;
*
sflu0 = sflu4 et sflu5;
sflu = sflu1 et sflu2 et sflu3 et sflu0;
elim tol sflu;
*
d1 = 0.05;
d2 = d1 * 2;

```

```

d3 = d2 * 2;
vtan0 = (sflu0 plus (ltan0 0 0)) volu dini d2 dfi d1
      tran ((0-ltan0) 0 0);
vtan1 = sflu volu dini d1 dfi d2 tran (ltan1 0 0);
vtanopen = (sflu plus (ltan1 0 0)) volu dini d2 dfi d3
      tran (ltanopen 0 0);
vtan2 = (sflu plus ((ltan1+ltanopen) 0 0)) volu dini d3 dfi d3
      tran (ltan2 0 0);
tank = vtan0 et vtan1 et vtanopen et vtan2;
elim tol tank;
*
pa1 = ltan1 rtank 0;
pa2 = (ltan1 + ltanopen) rtank 0;
cab = d pa1 pa2 dini d2 dfi d3;
abso = cab rota (2*nrpla) 90 p0 paxis;
elim tol (abso et tank);
*
tubehp = (sflu1 plus ((0 - ltube) 0 0)) volu dini d1 dfi d1
      tran (ltubhp 0 0);
tubelp = (sflu1 plus ((0 - ltublp) 0 0)) volu dini d1 dfi d1
      tran (ltublp 0 0);
tube = tubehp et tubelp;
*
flui = tube et tank;
elim tol flui;
trac cach tank;
trac cach flui;
*
list (nbel flui);
list (nbno flui);
list (nbel tube);
list (nbel tank);
list (nbel plat);
list (nbno plat);
*
mesh = flui et plat et abso;
tass mesh noop;
sauv form mesh;
*
fin;

```

## shoc52.epx

```

SHOC52
ECHO
!CONV win
CAST mesh
TRID ALE EROS 1.0
DIME ADAP NPOI 115000 CUVF 120000 NVFI 380000 ENDA
      NALE 1 NBLE 1 TERM
GEOM CUVF flui Q4GS plat CL3D abso TERM
COMP GROU 3 'endtube' LECT tube TERM COND XB GT -0.6
      'trigger' LECT tube TERM COND NEAR POIN -1.01 0 0
      'frefine' LECT flui TERM COND XB GT -1.2
      COND XB LT 1.0
      EPAI 0.8E-3 LECT plat _q4gs TERM
      COUL ROUG LECT tubehp TERM
      TURQ LECT tubelp tank DIFF trigger TERM
      VERT LECT plat TERM
      ROSE LECT trigger TERM
      JAUN LECT abso TERM
GRIL LAGR LECT plat TERM
EULE LECT flui TERM
MATE GAZP RO 59.459 GAMM 1.4 CV 719.286 PINI 50.E5 PREF 1.E5
      LECT tubehp TERM
      GAZP RO 1.189 GAMM 1.4 CV 719.286 PINI 1.E5 PREF 1.E5
      LECT tubelp tank _cuvf TERM
      VPJC RO 2700.0 YOUN 70.0E9 NU 0.3 ELAS 97.6E6 MXIT 200
      QR1 13.2E6 CR1 1458.0 QR2 10.6E6 CR2 185.9
      PDOT 5.E-4 C 1.4E-2 TQ 0.9 CP 910.0
      TM 893.0 M 1.0 DC 1.0 WC 44.6E6
      LECT plat _q4gs TERM
      CLVF ABSO RO 1.20
      LECT abso TERM
LINK COUP
      BLOQ 123456 LECT c2s c3s TERM
!
! cont spla not yet available in adaptivity for boundary-hanging nodes!
! use explicit blockages instead
!   CONT SPLA NX 0 NY 0 NZ 1 LECT c1s DIFF p1s TERM
!   CONT SPLA NX 0 NY 1 NZ 0 LECT c4s DIFF p3s TERM
      BLOQ 345 LECT c1s DIFF p0s p1s TERM
      BLOQ 246 LECT c4s DIFF p0s p3s TERM
      BLOQ 23456 LECT p0s TERM
LINK DECO FLSW STRU LECT plat TERM
      FLUI LECT flui TERM
      R 0.0436 ! 0.87*H_FLUID = 0.87*0.05 = 0.0435
      HGRI 0.06 ! Slightly bigger than structural element 0.05
      DGRI
      FACE
      BFLU 2 ! block if at least one node is in influence domain
      FSCP 1 ! couple in all directions
      ADAP LMAX 4 SCAL 16
ECRI DEPL VITE TFRE 1.E-3
      POIN LECT p0s TERM NOEL
      FICH SPLI ALIC FREQ 0 TFRE 0.DO
      TIME PROG 0.DO PAS 1.D-3 30.D-3 PAS 0.1D-3 40.D-3
      PAS 1.D-3 80.D-3
OPTI NOTE CSTA 0.5
      LOQ 1
      REZO GAMO 0.5

```

```

VFCC FCON 6 ! hllc solver
ORDR 2 ! order in space
OTPS 2 ! order in time
RECO 1
CALC TINI 0 TEND 23.E-3
FIN

```

## shoc52b.epx

```

Post treatment (visualization from alice file)
ECHO
RESU SPLI ALIC 'shoc52.ali' GARD PSCR
OPTI PRIN
SORT VISU NSTO 1
PLAY
CAME 6 EYE -1.00291E-01 -3.09784E+00 -3.09784E+00
! Q 3.82683E-01 9.23880E-01 0.00000E+00 0.00000E+00
VIEW 0.00000E+00 7.07107E-01 7.07107E-01
RIGH 1.00000E+00 0.00000E+00 0.00000E+00
UP 0.00000E+00 -7.07107E-01 7.07107E-01
FOV 2.48819E+01
!NAVIGATION MODE: ROTATING CAMERA
!CENTER : -1.00291E-01 4.00000E-01 4.00000E-01
!RSPHERE: 1.23667E+00
!RADIUS : 4.94669E+00
!ASPECT : 1.00000E+00
!NEAR : 3.71002E+00
!FAR : 7.42004E+00
CAME 5 EYE 5.61480E-02 -5.86912E-01 -5.86912E-01
! Q 3.82683E-01 9.23880E-01 0.00000E+00 0.00000E+00
VIEW 0.00000E+00 7.07107E-01 7.07107E-01
RIGH 1.00000E+00 0.00000E+00 0.00000E+00
UP 0.00000E+00 -7.07107E-01 7.07107E-01
FOV 2.48819E+01
!NAVIGATION MODE: ROTATING CAMERA
!CENTER : 5.61480E-02 7.50000E-02 7.50000E-02
!RSPHERE: 1.20011E-01
!RADIUS : 9.36084E-01
!ASPECT : 1.00000E+00
!NEAR : 8.16074E-01
!FAR : 1.17611E+00
SCEN GEOM NAVI FREE
LIMA ON
SLER CAM1 6 NFRA 1
TRAC OFFS FICH BMP OBJE LECT flui TERM REND
FREQ 23
GO
SCEN GEOM NAVI FREE
LINE HEOU
ISO FILL FIEL ECR0 1 SUPP LECT flui TERM
LIMA ON
SLER CAM1 6 NFRA 1
TRAC OFFS FICH BMP OBJE LECT flui TERM REND
ENDPLAY
FIN

```

## shoc52p.epx

```

Post-treatment (space curves from alice file)
ECHO
RESU SPLI ALIC 'shoc52.ali' GARD PSCR
COMP NGRO 1 'xaxo' LECT flui TERM
COND LINE X1 -18.2 Y1 0 Z1 0
X2 3.8 Y2 0 Z2 0 TOL 0.0001
SORT GRAP
PERF 'shoc52p.pun'
AXTE 1.0 'Time [s]'
SCOU 51 'p_21' NSTO 21 SAXE 1.0 'curr_abcissa' LECT xaxo TERM
SUPP LECT flui TERM
ECRO COMP 1
SCOU 52 'ro_21' NSTO 21 SAXE 1.0 'curr_abcissa' LECT xaxo TERM
SUPP LECT flui TERM
ECRO COMP 2
SCOU 55 'vx_21' NSTO 21 SAXE 1.0 'curr_abcissa' LECT xaxo TERM
SUPP LECT flui TERM
VCVI COMP 1
SCOU 61 'p_24' NSTO 24 SAXE 1.0 'curr_abcissa' LECT xaxo TERM
SUPP LECT flui TERM
ECRO COMP 1
SCOU 62 'ro_24' NSTO 24 SAXE 1.0 'curr_abcissa' LECT xaxo TERM
SUPP LECT flui TERM
ECRO COMP 2
SCOU 65 'vx_24' NSTO 24 SAXE 1.0 'curr_abcissa' LECT xaxo TERM
SUPP LECT flui TERM
VCVI COMP 1
TRAC 51 61 AXES 1.0 'PRESS. [PA]' XGRD YGRD
COLO ROUG NOIR
LIST 51 61 AXES 1.0 'PRESS. [PA]'
TRAC 52 62 AXES 1.0 'DENS. [KG/M3]' XGRD YGRD
COLO ROUG NOIR
LIST 52 62 AXES 1.0 'DENS. [KG/M3]'
TRAC 55 65 AXES 1.0 'VELOC. [M/S]' YZER XGRD YGRD
COLO ROUG NOIR
LIST 55 65 AXES 1.0 'VELOC. [M/S]'
FIN

```

## shoc52z.epx

```

Post treatment (visualization from alice file)
ECHO
RESU SPLI ALIC 'shoc51.ali' GARD PSCR
OPTI PRIN
SORT VISU NSTO 24
FIN

```

## shoc53.dgibi

```

opti echo 1;
opti dime 3 elem cub8;
opti trac psc ftra 'shoc53.mesh.ps';
opti sauv form 'shoc53.msh';
*
dpla = 0.3;
rpla = 0.5 * dpla;
dext = 0.625;
rext = 0.5 * dext;
dtank = 1.6;
rtank = 0.5 * dtank;
ltan0 = -0.7;
ltan1 = 0.9;
ltanopen = 1.4;
ltan2 = 1.5;
*
ltubhp = 2;
ltublp = 16.20;
ltube = ltubhp + ltublp;
*
nrpla = 3;
tol = 0.01 * dpla / nrpla;
*
p0s = 0 0 0;
p1s = 0 rpla 0;
p2s = 0 rpla rpla;
p3s = 0 0 rpla;
*
c1s = p0s d nrpla p1s;
c2s = p1s d nrpla p2s;
c3s = p2s d nrpla p3s;
c4s = p3s d nrpla p0s;
plat = dall c1s c2s c3s c4s plan;
*
p0 = p0s plus p0s;
p1 = p1s plus p0s;
p2 = p2s plus p0s;
p3 = p3s plus p0s;
*
c1 = p0 d nrpla p1;
c2 = p1 d nrpla p2;
c3 = p2 d nrpla p3;
c4 = p3 d nrpla p0;
sflui = dall c1 c2 c3 c4 plan;
*
paxis = 1 0 0;
p4 = 0 rext 0;
p5 = p4 tour 45 p0 paxis;
p6 = 0 0 rext;
*
c5 = p1 d nrpla p4;
c6 = p4 c nrpla p0 p5;
c7 = p5 d nrpla p2;
c8 = p2 d nrpla p1;
sflu2 = dall c5 c6 c7 c8 plan;
*
c9 = p2 d nrpla p5;
c10 = p5 c nrpla p0 p6;
c11 = p6 d nrpla p3;
c12 = p3 d nrpla p2;
sflu3 = dall c9 c10 c11 c12 plan;
*
p7 = 0 rtank 0;
p8 = p7 tour 45 p0 paxis;
p9 = 0 0 rtank;
*
din = (rext - rpla) / nrpla;
dfi = din * dtank / dext;
c13 = p4 d p7 dini din dfi dfi;
sflu4 = c13 rota 3 45 p0 paxis;
*
sflu5 = sflu4 tour 45 p0 paxis;
*
sflu0 = sflu4 et sflu5;
sflu = sflu1 et sflu2 et sflu3 et sflu0;
elim tol sflu;
*
d1 = 0.05;
d2 = d1 * 2;
d3 = d2 * 2;
vtan0 = (sflu0 plus (ltan0 0 0)) volu dini d2 dfi d1
tran ((0-ltan0) 0 0);
vtan1 = sflu volu dini d1 dfi d2 tran (ltan1 0 0);
vtanopen = (sflu plus (ltan1 0 0)) volu dini d2 dfi d3
tran (ltanopen 0 0);
vtan2 = (sflu plus ((ltan1+ltanopen) 0 0)) volu dini d3 dfi d3
tran (ltan2 0 0);
tank = vtan0 et vtan1 et vtanopen et vtan2;
elim tol tank;
*
pa1 = ltan1 rtank 0;
pa2 = (ltan1 + ltanopen) rtank 0;
cab = d pa1 pa2 dini d2 dfi d3;
abso = cab rota (2*nrpla) 90 p0 paxis;

```

```

elim tol (abso et tank);
*
tubehp = (sflui plus ((0 - ltube) 0 0)) volu dini d1 dfm d1
      tran (ltubhp 0 0);
tubelp = (sflui plus ((0 - ltublp) 0 0)) volu dini d1 dfm d1
      tran (ltublp 0 0);
tube = tubehp et tubelp;
*
flui = tube et tank;
elim tol flui;
trac cach tank;
trac cach flui;
*
list (nbcl flui);
list (nbno flui);
list (nbcl tube);
list (nbcl tank);
list (nbcl plat);
list (nbno plat);
*
mesh = flui et plat et abso;
tass mesh noop;
sauv form mesh;
*
fin;

```

## shoc53.epx

```

SHOC53
ECHO
!CONV win
CAST mesh
TRID ALE EROS 1.0
DIME ADAP NPOI 115000 CUVF 120000 NVFI 380000 ENDA
      NALE 1 NBLE 1 TERM
GEOM CUVF flui Q4GS plat CL3D abso TERM
COMP GROU 3 'endtube' LECT tube TERM COND XB GT -0.6
      'trigger' LECT tube TERM COND NEAR POIN -1.01 0 0
      'frefine' LECT flui TERM COND XB GT -1.2
      COND XB LT 1.0
      EPAI 0.8E-3 LECT plat _q4gs TERM
      COUL ROUG LECT tubehp TERM
      TURQ LECT tubelp tank DIFF trigger TERM
      VERT LECT plat TERM
      ROSE LECT trigger TERM
      JAUN LECT abso TERM
GRIL LAGR LECT plat TERM
      EULE LECT flui TERM
MATE GAZP RO 59.459 GAMM 1.4 CV 719.286 PINI 50.E5 PREF 1.E5
      LECT tubehp TERM
      GAZP RO 1.189 GAMM 1.4 CV 719.286 PINI 1.E5 PREF 1.E5
      LECT tubelp tank _cuvf TERM
      VPJC RO 2700.0 YOUN 70.0E9 NU 0.3 ELAS 97.6E6 MXIT 200
      QR1 13.2E6 CR1 1458.0 QR2 10.6E6 CR2 185.9
      PDOT 5.E-4 C 1.4E-2 TQ 0.9 CP 910.0
      TM 893.0 M 1.0 DC 1.0 WC 44.6E6
      LECT plat _q4gs TERM
      CLVF ABSO RO 1.20
      LECT abso TERM
LINK COUP
      BLOQ 123456 LECT c2s c3s TERM
!
! cont spla not yet available in adaptivity for boundary-hanging nodes!
! use explicit blockages instead
!
! CONT SPLA NX 0 NY 0 NZ 1 LECT c1s DIFF p1s TERM
! CONT SPLA NX 0 NY 1 NZ 0 LECT c4s DIFF p3s TERM
!
      BLOQ 345 LECT c1s DIFF p0s p1s TERM
      BLOQ 246 LECT c4s DIFF p0s p3s TERM
      BLOQ 23456 LECT p0s TERM
LINK DECO FLSW STRU LECT plat TERM
      FLUI LECT flui TERM
      R 0.0436 ! 0.87*H_FLUID = 0.87*0.05 = 0.0435
      HGRI 0.06 ! Slightly bigger than structural element 0.05
      DGRI
      FACE
      BFLU 2 ! block if at least one node is in influence domain
      FSCP 1 ! couple in all directions
      ADAP LMAX 4 SCAL 16
ECRI DEPL VITE TFRE 1.E-3
      POIN LECT p0s TERM NOEL
      FICH SPLI ALIC FREQ 0 TFRE 0.DO
      TIME PRDG 0.DO PAS 1.D-3 30.D-3 PAS 0.1D-3 40.D-3
      PAS 1.D-3 80.D-3
OPTI NOTE CSTA 0.5
      LOG 1
      REZO GAMO 0.5
      VFCC FCON 6 ! hllc solver
      ORDR 2 ! order in space
      OTPS 2 ! order in time
      RECO 1
      ADAP TRIG ECRO 1 TVAL 1.02E5 LECT trigger TERM
CALC TINI 0 TEND 23.E-3
FIN

```

## shoc53b.epx

```

Post treatment (visualization from alice file)
ECHO
RESU SPLI ALIC 'shoc53.ali' GARD PSCR

```

```

OPTI PRIN
SORT VISU NSTO 1
PLAY
CAME 6 EYE -1.00291E-01 -3.09784E+00 -3.09784E+00
! Q 3.82683E-01 9.23880E-01 0.00000E+00 0.00000E+00
      VIEW 0.00000E+00 7.07107E-01 7.07107E-01
      RIGH 1.00000E+00 0.00000E+00 0.00000E+00
      UP 0.00000E+00 -7.07107E-01 7.07107E-01
      FOV 2.48819E+01
!NAVIGATION MODE: ROTATING CAMERA
!CENTER : -1.00291E-01 4.00000E-01 4.00000E-01
!RSPHERE: 1.23667E+00
!RADIUS : 4.94669E+00
!ASPECT : 1.00000E+00
!NEAR : 3.71002E+00
!FAR : 7.42004E+00
CAME 5 EYE 5.61480E-02 -5.86912E-01 -5.86912E-01
! Q 3.82683E-01 9.23880E-01 0.00000E+00 0.00000E+00
      VIEW 0.00000E+00 7.07107E-01 7.07107E-01
      RIGH 1.00000E+00 0.00000E+00 0.00000E+00
      UP 0.00000E+00 -7.07107E-01 7.07107E-01
      FOV 2.48819E+01
!NAVIGATION MODE: ROTATING CAMERA
!CENTER : 5.61480E-02 7.50000E-02 7.50000E-02
!RSPHERE: 1.20011E-01
!RADIUS : 9.36084E-01
!ASPECT : 1.00000E+00
!NEAR : 8.16074E-01
!FAR : 1.17611E+00
SCEN GEOM NAVI FREE
      LIMA ON
      SLER CAM1 6 NFRA 1
      TRAC OFFS FICH BMP OBJE LECT flui TERM REND
      FREQ 23
      GO
      SCEN GEOM NAVI FREE
      LINE HEOU
      ISO FILL FIEL ECRO 1 SUPP LECT flui TERM
      LIMA ON
      SLER CAM1 6 NFRA 1
      TRAC OFFS FICH BMP OBJE LECT flui TERM REND
      ENDPLAY
      FIN

```

## shoc53p.epx

```

Post-treatment (space curves from alice file)
ECHO
RESU SPLI ALIC 'shoc53.ali' GARD PSCR
COMP NGRO 1 'xaxo' LECT flui TERM
      COND LINE X1 -18.2 Y1 0 Z1 0
      X2 3.8 Y2 0 Z2 0 TOL 0.0001
SORT GRAP
PERF 'shoc53p.pun'
AXTE 1.0 'Time [s]'
SCOU 51 'p_21' NSTO 21 SAXE 1.0 'curr_abcissa' LECT xaxo TERM
      SUPP LECT flui TERM
      ECRO COMP 1
SCOU 52 'ro_21' NSTO 21 SAXE 1.0 'curr_abcissa' LECT xaxo TERM
      SUPP LECT flui TERM
      ECRO COMP 2
SCOU 55 'vx_21' NSTO 21 SAXE 1.0 'curr_abcissa' LECT xaxo TERM
      SUPP LECT flui TERM
      VCVI COMP 1
SCOU 61 'p_24' NSTO 24 SAXE 1.0 'curr_abcissa' LECT xaxo TERM
      SUPP LECT flui TERM
      ECRO COMP 1
SCOU 62 'ro_24' NSTO 24 SAXE 1.0 'curr_abcissa' LECT xaxo TERM
      SUPP LECT flui TERM
      ECRO COMP 2
SCOU 65 'vx_24' NSTO 24 SAXE 1.0 'curr_abcissa' LECT xaxo TERM
      SUPP LECT flui TERM
      VCVI COMP 1
TRAC 51 61 AXES 1.0 'PRESS. [PA]' XGRD YGRD
      COLO ROUG NOIR
      LIST 51 61 AXES 1.0 'PRESS. [PA]'
      TRAC 52 62 AXES 1.0 'DENS. [KG/M3]' XGRD YGRD
      COLO ROUG NOIR
      LIST 52 62 AXES 1.0 'DENS. [KG/M3]'
      TRAC 55 65 AXES 1.0 'VELOC. [M/S]' YZER XGRD YGRD
      COLO ROUG NOIR
      LIST 55 65 AXES 1.0 'VELOC. [M/S]'
      FIN

```

## shoc54.dgibi

```

opti echo 1;
opti dime 3 elem cub8;
opti trac psc ftra 'shoc54_mesh.ps';
opti sauv form 'shoc54.msh';
*
dpla = 0.3;
rpla = 0.5 * dpla;
dext = 0.625;
rext = 0.5 * dext;
dtank = 1.6;
rtank = 0.5 * dtank;
ltan0 = -0.7;
ltan1 = 0.9;
ltanopen = 1.4;
ltan2 = 1.5;
*

```

```

ltubhp = 2;
ltublp = 16.20;
ltube = ltubhp + ltublp;
*
nrpla = 3;
tol = 0.01 * dpla / nrpla;
*
p0s = 0 0 0;
p1s = 0 rpla 0;
p2s = 0 rpla rpla;
p3s = 0 0 rpla;
*
c1s = p0s d nrpla p1s;
c2s = p1s d nrpla p2s;
c3s = p2s d nrpla p3s;
c4s = p3s d nrpla p0s;
plat = dall c1s c2s c3s c4s plan;
*
p0 = p0s plus p0s;
p1 = p1s plus p0s;
p2 = p2s plus p0s;
p3 = p3s plus p0s;
*
c1 = p0 d nrpla p1;
c2 = p1 d nrpla p2;
c3 = p2 d nrpla p3;
c4 = p3 d nrpla p0;
sflu1 = dall c1 c2 c3 c4 plan;
*
paxis = 1 0 0;
p4 = 0 rext 0;
p5 = p4 tour 45 p0 paxis;
p6 = 0 0 rext;
*
c5 = p1 d nrpla p4;
c6 = p4 c nrpla p0 p5;
c7 = p5 d nrpla p2;
c8 = p2 d nrpla p1;
sflu2 = dall c5 c6 c7 c8 plan;
*
c9 = p2 d nrpla p5;
c10 = p5 c nrpla p0 p6;
c11 = p6 d nrpla p3;
c12 = p3 d nrpla p2;
sflu3 = dall c9 c10 c11 c12 plan;
*
p7 = 0 rtank 0;
p8 = p7 tour 45 p0 paxis;
p9 = 0 0 rtank;
*
din = (rext - rpla) / nrpla;
dfi = din * dtank / dext;
c13 = p4 d p7 dini din dfi;
sflu4 = c13 rota 3 45 p0 paxis;
*
sflu5 = sflu4 tour 45 p0 paxis;
*
sflu0 = sflu4 et sflu5;
sflu = sflu1 et sflu2 et sflu3 et sflu0;
elim tol sflu;
*
d1 = 0.05;
d2 = d1 * 2;
d3 = d2 * 2;
vtan0 = (sflu0 plus ((0 - ltube) 0 0)) volu dini d2 dfin d1
      tran ((0-ltan0) 0 0);
vtan1 = sflu volu dini d1 dfin d2 tran (ltan1 0 0);
vtanopen = (sflu plus (ltan1 0 0)) volu dini d2 dfin d3
      tran (ltanopen 0 0);
vtan2 = (sflu plus ((ltan1+ltanopen) 0 0)) volu dini d3 dfin d3
      tran (ltan2 0 0);
tank = vtan0 et vtan1 et vtanopen et vtan2;
elim tol tank;
*
pa1 = ltan1 rtank 0;
pa2 = (ltan1 + ltanopen) rtank 0;
cab = d pa1 pa2 dini d2 dfin d3;
abso = cab rota (2*nrpla) 90 p0 paxis;
elim tol (abso et tank);
*
tubehp = (sflu1 plus ((0 - ltube) 0 0)) volu dini d1 dfin d1
      tran (ltubhp 0 0);
tubelp = (sflu1 plus ((0 - ltublp) 0 0)) volu dini d1 dfin d1
      tran (ltublp 0 0);
tube = tubehp et tubelp;
*
flui = tube et tank;
elim tol flui;
trac cach tank;
trac cach flui;
*
list (nbcl flui);
list (nbno flui);
list (nbcl tube);
list (nbcl tank);
list (nbcl plat);
list (nbno plat);
*
mesh = flui et plat et abso;
tass mesh noop;
sauv form mesh;
*
fin;

```

## shoc54.epx

```

SHOC54
ECHO
!CONV win
CAST mesh
TRID ALE EROS 1.0
DIME ADAP NPOI 115000 CUVF 120000 NVFI 380000 ENDA
      NALE 1 NBLE 1 TERM
GEOM CUVF flui Q4GS plat CL3D abso TERM
COMP GROU 3 'endtube' LECT tube TERM COND XB GT -0.6
      'trigger' LECT tube TERM COND NEAR POIN -1.01 0 0
      'frefine' LECT flui TERM COND XB GT -1.2
      COND XB LT 1.0
      EPAI 0.8E-3 LECT plat _q4gs TERM
      COUL ROUG LECT tubehp TERM
      TURQ LECT tubelp tank DIFF trigger TERM
      VERT LECT plat TERM
      ROSE LECT trigger TERM
      JAUN LECT abso TERM
GRIL LAGR LECT plat TERM
      EULE LECT flui TERM
MATE GAZP RO 59.459 GAMM 1.4 CV 719.286 PINI 50.E5 PREF 1.E5
      LECT tubehp TERM
      GAZP RO 1.189 GAMM 1.4 CV 719.286 PINI 1.E5 PREF 1.E5
      LECT tubelp tank _cuvf TERM
VPJC RO 2700.0 YOUN 70.OE9 NU 0.3 ELAS 97.6E6 MXIT 200
      QR1 13.2E6 CR1 1458.0 QR2 10.6E6 CR2 185.9
      PDOT 5.E-4 C 1.4E-2 TQ 0.9 CP 910.0
      TM 893.0 M 1.0 DC 1.0 WC 44.6E6
      LECT plat _q4gs TERM
      CLVF ABSO RO 1.20
      LECT abso TERM
LINK COUP
      BLOQ 123456 LECT c2s c3s TERM
!
! cont spla not yet available in adaptivity for boundary-hanging nodes!
! use explicit blockages instead
! CONT SPLA NX 0 NY 0 NZ 1 LECT c1s DIFF p1s TERM
! CONT SPLA NX 0 NY 1 NZ 0 LECT c4s DIFF p3s TERM
      BLOQ 345 LECT c1s DIFF p0s p1s TERM
      BLOQ 246 LECT c4s DIFF p0s p3s TERM
      BLOQ 23456 LECT p0s TERM
LINK DECO FLSW STRU LECT plat TERM
      FLUI LECT flui TERM
      R 0.0436 ! 0.87*H_FLUID = 0.87*0.05 = 0.0435
      HGRI 0.06 ! Slightly bigger than structural element 0.05
      DGRI
      FACE
      BFLU 2 ! block if at least one node is in influence domain
      FSCP 1 ! couple in all directions
      ADAP LMAX 4 SCAL 16
ECRI DEPL VITE TFRE 1.E-3
      POIN LECT p0s TERM NOEL
      FICH SPLI ALIC FREQ 0 TFRE 0.DO
      TIME PROG 0.DO PAS 1.D-3 30.D-3 PAS 0.1D-3 40.D-3
      PAS 1.D-3 80.D-3
OPTI NOTE CSTA 0.5
      LQ 1
      REZO GAMO 0.5
      VFCC FC0N 6 ! hllc solver
      ORDR 2 ! order in space
      OTPS 2 ! order in time
      RECO 1
      ADAP TRIG ECRO 1 TVAL 1.02E5 LECT trigger TERM
      NOCR TRIG LECT plat TERM
CALC TINI 0 TEND 23.E-3
FIN

```

## shoc55.dgibi

```

opti echo 1;
opti dime 3 elem cub8;
opti trac psc ftra 'shoc55_mesh.ps';
opti sauv form 'shoc55.msh';
*
dpla = 0.3;
rpla = 0.5 * dpla;
dext = 0.625;
rext = 0.5 * dext;
dtank = 1.6;
rtank = 0.5 * dtank;
ltan0 = -0.7;
ltan1 = 0.9;
ltanopen = 1.4;
ltan2 = 1.5;
*
ltubhp = 2;
ltublp = 16.20;
ltube = ltubhp + ltublp;
*
nrpla = 3;
tol = 0.01 * dpla / nrpla;
*
p0s = 0 0 0;
p1s = 0 rpla 0;
p2s = 0 rpla rpla;
p3s = 0 0 rpla;
*
c1s = p0s d nrpla p1s;
c2s = p1s d nrpla p2s;

```

```

c3s = p2s d nrpla p3s;
c4s = p3s d nrpla p0s;
plat = dall c1s c2s c3s c4s plan;
*
p0 = p0s plus p0s;
p1 = p1s plus p0s;
p2 = p2s plus p0s;
p3 = p3s plus p0s;
*
c1 = p0 d nrpla p1;
c2 = p1 d nrpla p2;
c3 = p2 d nrpla p3;
c4 = p3 d nrpla p0;
sflu1 = dall c1 c2 c3 c4 plan;
*
paxis = 1 0 0;
p4 = 0 rext 0;
p5 = p4 tour 45 p0 paxis;
p6 = 0 0 rext;
*
c5 = p1 d nrpla p4;
c6 = p4 c nrpla p0 p5;
c7 = p5 d nrpla p2;
c8 = p2 d nrpla p1;
sflu2 = dall c5 c6 c7 c8 plan;
*
c9 = p2 d nrpla p5;
c10 = p5 c nrpla p0 p6;
c11 = p6 d nrpla p3;
c12 = p3 d nrpla p2;
sflu3 = dall c9 c10 c11 c12 plan;
*
p7 = 0 rtank 0;
p8 = p7 tour 45 p0 paxis;
p9 = 0 0 rtank;
*
din = (rext - rpla) / nrpla;
dfi = din * dtank / dext;
c13 = p4 d p7 dini din dfi;
sflu4 = c13 rota 3 45 p0 paxis;
*
sflu5 = sflu4 tour 45 p0 paxis;
*
sflu0 = sflu4 et sflu5;
sflu = sflu1 et sflu2 et sflu3 et sflu0;
elim tol sflu;
*
d1 = 0.05;
d2 = d1 * 2;
d3 = d2 * 2;
vtan0 = (sflu0 plus (ltan0 0 0)) volu dini d2 dfi d1
      tran ((0-ltan0) 0 0);
vtan1 = sflu volu dini d1 dfi d2 tran (ltan1 0 0);
vtanopen = (sflu plus (ltan1 0 0)) volu dini d2 dfi d3
      tran (ltanopen 0 0);
vtan2 = (sflu plus ((ltan1+ltanopen) 0 0)) volu dini d3 dfi d3
      tran (ltan2 0 0);
tank = vtan0 et vtan1 et vtanopen et vtan2;
elim tol tank;
*
pa1 = ltan1 rtank 0;
pa2 = (ltan1 + ltanopen) rtank 0;
cab = d pa1 pa2 dini d2 dfi d3;
abso = cab rota (2*nrpla) 90 p0 paxis;
elim tol (abso et tank);
*
tubehp = (sflu1 plus ((0 - ltube) 0 0)) volu dini d1 dfi d1
      tran (ltubhp 0 0);
tubelp = (sflu1 plus ((0 - ltublp) 0 0)) volu dini d1 dfi d1
      tran (ltublp 0 0);
tube = tubehp et tubelp;
*
flui = tube et tank;
elim tol flui;
trac cach tank;
trac cach flui;
*
list (nbel flui);
list (nbno flui);
list (nbel tube);
list (nbel tank);
list (nbel plat);
list (nbno plat);
*
mesh = flui et plat et abso;
tass mesh noop;
sau form mesh;
*
fin;

```

## shoc55.epx

```

SHOC55
ECHO
!CONV win
CAST mesh
TRID ALE EROS 1.0
DIME ADAP NPOI 115000 CUVF 120000 NVFI 380000 ENDA
NALE 1 NBLE 1 TERM
GEOM CUVF flui Q4GS plat CL3D abso TERM
COMP GROU 3 'endtube' LECT tube TERM COND XB GT -0.6

```

```

'trigger' LECT tube TERM COND NEAR POIN -1.01 0 0
'frefine' LECT flui TERM COND XB GT -1.2
COND XB LT 1.0
EPAI 0.8E-3 LECT plat _q4gs TERM
COUL ROUG LECT tubehp TERM
TURQ LECT tubelp tank DIFF trigger TERM
VERT LECT plat TERM
ROSE LECT trigger TERM
JAUN LECT abso TERM
GRIL LAGR LECT plat TERM
EULE LECT flui TERM
MATE GAZP RO 59.459 GAMM 1.4 CV 719.286 PINI 50.E5 PREF 1.E5
LECT tubehp TERM
GAZP RO 1.189 GAMM 1.4 CV 719.286 PINI 1.E5 PREF 1.E5
LECT tubelp tank _cuvf TERM
VPJC RO 2700.0 YOUN 70.0E9 NU 0.3 ELAS 97.6E6 MXIT 200
QR1 13.2E6 CR1 1458.0 QR2 10.6E6 CR2 185.9
PDOT 5.E-4 C 1.4E-2 TQ 0.9 CP 910.0
TM 893.0 M 1.0 DC 1.0 WC 44.6E6
LECT plat _q4gs TERM
CLVF ABSO RO 1.20
LECT abso TERM
LINK COUP
BLOQ 123456 LECT c2s c3s TERM
!
! cont spla not yet available in adaptivity for boundary-hanging nodes!
! use explicit blockages instead
! CONT SPLA NX 0 NY 0 NZ 1 LECT c1s DIFF p1s TERM
! CONT SPLA NX 0 NY 1 NZ 0 LECT c4s DIFF p3s TERM
BLOQ 345 LECT c1s DIFF p0s p1s TERM
BLOQ 246 LECT c4s DIFF p0s p3s TERM
BLOQ 23456 LECT p0s TERM
TBLO 126 TRIG LECT c1s DIFF p0s p1s TERM
135 TRIG LECT c4s DIFF p0s p3s TERM
1 TRIG LECT p0s TERM
123456 TRIG LECT plat DIFF c1s c2s c3s c4s TERM
LINK DECO FLSW STRU LECT plat TERM
FLUI LECT flui TERM
R 0.0436 ! 0.87*H_FLUID = 0.87*0.05 = 0.0435
HGRI 0.06 ! Slightly bigger than structural element 0.05
DGRI
FACE
BFLU 2 ! block if at least one node is in influence domain
FSCP 1 ! couple in all directions
ADAP LMAX 4 SCAL 16
ECRI DEPL VITE TFRE 1.E-3
POIN LECT p0s TERM NOEL
FICH SPLI ALIC FREQ 0 TFRE 0.D0
TIME PROG 0.D0 PAS 1.D-3 30.D-3 PAS 0.1D-3 40.D-3
PAS 1.D-3 80.D-3
OPTI NOTE CSTA 0.5
LOG 1
REZO GAMO 0.5
VFCC FCON 6 ! hllc solver
ORDR 2 ! order in space
OTPS 2 ! order in time
RECO 1
ADAP TRIG ECRO 1 TVAL 1.02E5 LECT trigger TERM
NOCR TRIG LECT plat TERM
CALC TINI 0 TEND 23.E-3
FIN

```

## shoc55b.epx

```

Post treatment (visualization from alice file)
ECHO
RESU SPLI ALIC 'shoc55.ali' GARD PSCR
OPTI PRIN
SORT VISU NSTO 1
PLAY
CAME 6 EYE -1.00291E-01 -3.09784E+00 -3.09784E+00
! Q 3.82683E-01 9.23880E-01 0.00000E+00 0.00000E+00
VIEW 0.00000E+00 7.07107E-01 7.07107E-01
RIGH 1.00000E+00 0.00000E+00 0.00000E+00
UP 0.00000E+00 -7.07107E-01 7.07107E-01
FOV 2.48819E+01
!NAVIGATION MODE: ROTATING CAMERA
!CENTER : -1.00291E-01 4.00000E-01 4.00000E-01
!RSPHERE: 1.23667E+00
!RADIUS : 4.94669E+00
!ASPECT : 1.00000E+00
!NEAR : 3.71002E+00
!FAR : 7.42004E+00
CAME 5 EYE 5.61480E-02 -5.86912E-01 -5.86912E-01
! Q 3.82683E-01 9.23880E-01 0.00000E+00 0.00000E+00
VIEW 0.00000E+00 7.07107E-01 7.07107E-01
RIGH 1.00000E+00 0.00000E+00 0.00000E+00
UP 0.00000E+00 -7.07107E-01 7.07107E-01
FOV 2.48819E+01
!NAVIGATION MODE: ROTATING CAMERA
!CENTER : 5.61480E-02 7.50000E-02 7.50000E-02
!RSPHERE: 1.20011E-01
!RADIUS : 9.36084E-01
!ASPECT : 1.00000E+00
!NEAR : 8.16074E-01
!FAR : 1.17611E+00
SCEN GEOM NAVI FREE
LIMA ON
SLER CAM1 6 NFRA 1
TRAC OFFS FICH BMP OBJE LECT flui TERM REND
FREQ 23

```

```

GO
SCEN GEOM NAVI FREE
      LINE HEOU
      ISO FILL FIEL ECRO 1 SUPP LECT flui TERM
      LIMA ON
SLER CAM1 6 NFRA 1
TRAC OFFS FICH BMP OBJE LECT flui TERM REND
ENDPLAY
FIN

```

## shoc55p.epx

```

Post-treatment (space curves from alice file)
ECHO
RESU SPLI ALIC 'shoc55.ali' GARD PSCR
COMP NGRO 1 'xaxo' LECT flui TERM
      COND LINE X1 -18.2 Y1 0 Z1 0
      X2 3.8 Y2 0 Z2 0 TOL 0.0001

SORT GRAP
PERF 'shoc55p.pun'
AXTE 1.0 'Time [s]'
SCOU 51 'p_21' NSTO 21 SAXE 1.0 'curr_abcissa' LECT xaxo TERM
      SUPP LECT flui TERM
      ECRO COMP 1
SCOU 52 'ro_21' NSTO 21 SAXE 1.0 'curr_abcissa' LECT xaxo TERM
      SUPP LECT flui TERM
      ECRO COMP 2
SCOU 55 'vx_21' NSTO 21 SAXE 1.0 'curr_abcissa' LECT xaxo TERM
      SUPP LECT flui TERM
      VCVI COMP 1
SCOU 61 'p_24' NSTO 24 SAXE 1.0 'curr_abcissa' LECT xaxo TERM
      SUPP LECT flui TERM
      ECRO COMP 1
SCOU 62 'ro_24' NSTO 24 SAXE 1.0 'curr_abcissa' LECT xaxo TERM
      SUPP LECT flui TERM
      ECRO COMP 2
SCOU 65 'vx_24' NSTO 24 SAXE 1.0 'curr_abcissa' LECT xaxo TERM
      SUPP LECT flui TERM
      VCVI COMP 1
TRAC 51 61 AXES 1.0 'PRESS. [PA]' XGRD YGRD
COLO ROUG NOIR
LIST 51 61 AXES 1.0 'PRESS. [PA]'
TRAC 52 62 AXES 1.0 'DENS. [KG/M3]' XGRD YGRD
COLO ROUG NOIR
LIST 52 62 AXES 1.0 'DENS. [KG/M3]'
TRAC 55 65 AXES 1.0 'VELOC. [M/S]' YZER XGRD YGRD
COLO ROUG NOIR
LIST 55 65 AXES 1.0 'VELOC. [M/S]'
FIN

```

## shoc55z.epx

```

Post treatment (visualization from alice file)
ECHO
RESU SPLI ALIC 'shoc55.ali' GARD PSCR
OPTI PRIN
SORT VISU NSTO 24
FIN

```

## shoc61.dgibi

```

opti echo 1;
opti dime 3 elem cub8;
opti trac psc ftra 'shoc61.mesh.ps';
opti sauv form 'shoc61.msh';
*
dpla = 0.3;
rpla = 0.5 * dpla;
dext = 0.625;
rxt = 0.5 * dext;
dtank = 1.6;
rtank = 0.5 * dtank;
ltan0 = -0.7;
ltan1 = 0.9;
ltanopen = 1.4;
ltan2 = 1.5;
*
ltubhp = 2;
ltublp = 16.20;
ltube = ltubhp + ltublp;
*
nrpla = 3;
tol = 0.01 * dpla / nrpla;
*
p0s = 0 0 0;
p1s = 0 rpla 0;
p2s = 0 rpla rpla;
p3s = 0 0 rpla;
*
c1s = p0s d nrpla p1s;
c2s = p1s d nrpla p2s;
c3s = p2s d nrpla p3s;
c4s = p3s d nrpla p0s;
plat = dall c1s c2s c3s c4s plan;
*
p0 = p0s plus p0s;
p1 = p1s plus p0s;

```

```

p2 = p2s plus p0s;
p3 = p3s plus p0s;
*
c1 = p0 d nrpla p1;
c2 = p1 d nrpla p2;
c3 = p2 d nrpla p3;
c4 = p3 d nrpla p0;
sflu1 = dall c1 c2 c3 c4 plan;
*
paxis = 1 0 0;
p4 = 0 rxt 0;
p5 = p4 tour 45 p0 paxis;
p6 = 0 0 rxt;
*
c5 = p1 d nrpla p4;
c6 = p4 c nrpla p0 p5;
c7 = p5 d nrpla p2;
c8 = p2 d nrpla p1;
sflu2 = dall c5 c6 c7 c8 plan;
*
c9 = p2 d nrpla p5;
c10 = p5 c nrpla p0 p6;
c11 = p6 d nrpla p3;
c12 = p3 d nrpla p2;
sflu3 = dall c9 c10 c11 c12 plan;
*
p7 = 0 rtank 0;
p8 = p7 tour 45 p0 paxis;
p9 = 0 0 rtank;
*
din = (rxt - rpla) / nrpla;
dfi = din * dtank / dext;
c13 = p4 d p7 dini din dfi dfi;
sflu4 = c13 rota 3 45 p0 paxis;
*
sflu5 = sflu4 tour 45 p0 paxis;
*
sflu0 = sflu4 et sflu5;
sflu = sflu1 et sflu2 et sflu3 et sflu0;
elim tol sflu;
*
d1 = 0.05;
d2 = d1 * 2;
d3 = d2 * 2;
vtan0 = (sflu0 plus (ltan0 0 0)) volu dini d2 dfi d1
      tran ((0-ltan0) 0 0);
vtan1 = sflu volu dini d1 dfi d2 tran (ltan1 0 0);
vtanopen = (sflu plus (ltan1 0 0)) volu dini d2 dfi d3
      tran (ltanopen 0 0);
vtan2 = (sflu plus ((ltan1+ltanopen) 0 0)) volu dini d3 dfi d3
      tran (ltan2 0 0);
tank = vtan0 et vtan1 et vtanopen et vtan2;
elim tol tank;
*
pa1 = ltan1 rtank 0;
pa2 = (ltan1 + ltanopen) rtank 0;
cab = d pa1 pa2 dini d2 dfi d3;
abso = cab rota (2*nrpla) 90 p0 paxis;
elim tol (abso et tank);
*
*tubehp = (sflu1 plus ((0 - ltube) 0 0)) volu dini d1 dfi d1
      tran (ltubhp 0 0);
*tubelp = (sflu1 plus ((0 - ltublp) 0 0)) volu dini d1 dfi d1
      tran (ltublp 0 0);
pid1 = (0 - ltube) 0 0;
pid2 = pid1 plus (ltubhp 0 0);
tubehp = pid1 d pid2 dini d1 dfi d1;
lenlp3d = 1.2;
pid3 = (0 - lenlp3d) 0 0;
tubelp1 = pid2 d pid3 dini d1 dfi d1;
stub3d = sflu1 plus pid3;
tubelp3 = stub3d volu dini d1 dfi d1 tran (lenlp3d 0 0);
tubelp = tubelp1 et tubelp3;
*
raclp = manu supe (pid3 et stub3d);
*
tube = tubehp et tubelp;
*
flui = tube et tank;
elim tol flui;
trac cach tank;
trac cach flui;
*
list (nbel flui);
list (nbno flui);
list (nbel tube);
list (nbel tank);
list (nbel plat);
list (nbno plat);
*
mesh = flui et plat et abso et raclp;
tass mesh noop;
sauv form mesh;
*
fin;

```

## shoc61.epx

```

SHOC61
ECHO
!CONV win

```



```

CAST mesh
TRID ALE EROS 1.0
DIME JONC 17 ! Total n. of nodes in a TUBM junction
      NALE 1 NBLE 1 TERM
GEOM CUVF tubelp3 tank TUVF tubehp tubelp1 Q4GS plat
! Attention: we are obliged to assign (dummy) CL3D elements to the stub3d
! object (3D part of the 1D-3D junction), because the name of this object
! is needed below in the RACC TUBM FACE directive and it seems that the
! code expects there an object made of elements, not just of nodes
! (although only the nodes are really needed, at least according
! to the manual)
      CL3D abso stub3d
      TUBM raclp TERM
COMP DIAM DROI 0.1692568 LECT tubehp tubelp1 TERM
      RACC TUBM LECT raclp TERM
      NTUB LECT pid3 TERM DTUB 0.1692568
      FACE LECT stub3d TERM COEF 1.0
      EPAI 0.8E-3 LECT plat TERM
! Attention: the TUBM element (raclp) is NOT included
! in the "mesh" object (although it IS indeed passed in from Cast3m).
! For this reason we must add it explicitly in the GROU directive below
! if we want to have it in the extracted element groups.
      GROU 4 'endtube' LECT tube TERM COND XB GT -0.6
      'trigger' LECT tube TERM COND NEAR POIN -1.01 0 0
      'frefine' LECT flui TERM COND XB GT -1.2
      COND XB LT 1.0
      'nraclp' LECT mesh raclp TERM
      COND XB GT -1.201 COND XB LT -1.199
      COUL ROUG LECT tubehp TERM
      TURQ LECT tubelp tank DIFF trigger TERM
      VERT LECT plat TERM
      ROSE LECT trigger TERM
      JAUN LECT abso TERM
      BLEU LECT raclp TERM
GRIL LAGR LECT plat TERM
      EULE LECT flui TERM
MATE GAZP RO 59.459 GAMM 1.4 CV 719.286 PINI 50.E5 PREF 1.E5
      LECT tubehp TERM
      GAZP RO 1.189 GAMM 1.4 CV 719.286 PINI 1.E5 PREF 1.E5
      LECT tubelp1 raclp TERM
! In order to obtain a printout at least of the 3D VFCCs I am obliged
! to use a different material for tubelp3 and other 3D parts, but with the
! same characteristics as the material used for tubelp1
      GAZP RO 1.189 GAMM 1.4 CV 719.286 PINI 1.E5 PREF 1.E5
      LECT tubelp3 tank TERM
      VPJC RO 2700.0 YOUN 70.0E9 NU 0.3 ELAS 97.6E6 MXIT 200
      QR1 13.2E6 CR1 1458.0 QR2 10.6E6 CR2 185.9
      PDOT 5.E-4 C 1.4E-2 TQ 0.9 CP 910.0
      TM 893.0 M 1.0 DC 1.0 WC 44.6E6
      LECT plat TERM
      CLVF ABDO RO 1.20
      LECT abso TERM
LINK COUP
      BLOQ 123456 LECT c2s c3s TERM
!
! cont spla not yet available in adaptivity for boundary-hanging nodes!
! use explicit blockages instead
!   CONT SPLA NX 0 NY 0 NZ 1 LECT c1s DIFF p1s TERM
!   CONT SPLA NX 0 NY 1 NZ 0 LECT c4s DIFF p3s TERM
      BLOQ 345 LECT c1s DIFF p0s p1s TERM
      BLOQ 246 LECT c4s DIFF p0s p3s TERM
      BLOQ 23456 LECT p0s TERM
LINK DECO FLSW STRU LECT plat TERM
      FLUI LECT flui TERM
      R 0.0436 ! 0.87*H_FLUID = 0.87*0.05 = 0.0435
      HGRI 0.06 ! Slightly bigger than structural element 0.05
      DGRI
      FACE
      BFLU 2 ! block if at least one node is in influence domain
      FSCP 1 ! couple in all directions
ECRI DEPL VITE TFRE 1.E-3
      POIN LECT p0s TERM NOEL
      FICH SPLI ALIC FREQ 0 TFRE 0.DO
      TIME PROG 0.DO PAS 1.D-3 30.D-3 PAS 0.1D-3 40.D-3
      PAS 1.D-3 80.D-3
OPTI NOTE CSTA 0.5
      LOG 1
      REZO GAMO 0.5
      VFCC FC0N 6 ! hllc solver
      ORDR 2 ! order in space
      OTPS 2 ! order in time
      RECO 1 ! Not accepted by CAL_VFCC_1D
CALC TINI 0 TEND 23.E-3
FIN

```

## shoc61b.epx

```

Post treatment (visualization from alice file)
ECHO
RESU SPLI ALIC 'shoc61.ali' GARD PSCR
OPTI PRIN
SORT VISU NSTO 24
PLAY
CAME 6 EYE -1.00291E-01 -3.09784E+00 -3.09784E+00
! Q 3.82683E-01 9.23880E-01 0.00000E+00 0.00000E+00
VIEW 0.00000E+00 7.07107E-01 7.07107E-01
RIGH 1.00000E+00 0.00000E+00 0.00000E+00
UP 0.00000E+00 -7.07107E-01 7.07107E-01
FOV 2.48819E+01
!NAVIGATION MODE: ROTATING CAMERA
!CENTER : -1.00291E-01 4.00000E-01 4.00000E-01

```

```

!RSPHERE: 1.23667E+00
!RADIUS : 4.94669E+00
!ASPECT : 1.00000E+00
!NEAR : 3.71002E+00
!FAR : 7.42004E+00
CAME 5 EYE 5.61480E-02 -5.86912E-01 -5.86912E-01
! Q 3.82683E-01 9.23880E-01 0.00000E+00 0.00000E+00
VIEW 0.00000E+00 7.07107E-01 7.07107E-01
RIGH 1.00000E+00 0.00000E+00 0.00000E+00
UP 0.00000E+00 -7.07107E-01 7.07107E-01
FOV 2.48819E+01
!NAVIGATION MODE: ROTATING CAMERA
!CENTER : 5.61480E-02 7.50000E-02 7.50000E-02
!RSPHERE: 1.20011E-01
!RADIUS : 9.36084E-01
!ASPECT : 1.00000E+00
!NEAR : 8.16074E-01
!FAR : 1.17611E+00
SCEN GEOM NAVI FREE
ISO FILL FIEL ECRO 1 SUPP LECT flui TERM
LIMA ON
SLER CAM1 6 NFRA 1
TRAC OFFS FICH BMP OBJE LECT flui TERM REND
SCEN OBJE USLM LECT flui TERM
GEOM NAVI FREE
FACE SBAC
LINE SSHA SFRE
VECT SCCO FIEL VITE SUPP LECT plat TERM
LENG 0.1
LIMA ON
SLER CAM1 5 NFRA 1
TRAC OFFS FICH BMP OBJE LECT flui plat TERM NFAI REND
ENDPLAY
FIN

```

## shoc61p.epx

```

Post-treatment (space curves from alice file)
ECHO
RESU SPLI ALIC 'shoc61.ali' GARD PSCR
COMP NGRO 1 'xaxo' LECT flui TERM
      COND LINE X1 -18.2 Y1 0 Z1 0
      X2 3.8 Y2 0 Z2 0 TOL 0.0001
SORT GRAP
PERF 'shoc61p.pun'
AXTE 1.0 'Time [s]'
SCOU 51 'p_21' NSTO 21 SAXE 1.0 'curr_abcissa' LECT xaxo TERM
      ECRO COMP 1
SCOU 52 'ro_21' NSTO 21 SAXE 1.0 'curr_abcissa' LECT xaxo TERM
      ECRO COMP 2
SCOU 55 'vx_21' NSTO 21 SAXE 1.0 'curr_abcissa' LECT xaxo TERM
      VCVI COMP 1
SCOU 61 'p_24' NSTO 24 SAXE 1.0 'curr_abcissa' LECT xaxo TERM
      ECRO COMP 1
SCOU 62 'ro_24' NSTO 24 SAXE 1.0 'curr_abcissa' LECT xaxo TERM
      ECRO COMP 2
SCOU 65 'vx_24' NSTO 24 SAXE 1.0 'curr_abcissa' LECT xaxo TERM
      VCVI COMP 1
TRAC 51 61 AXES 1.0 'PRESS. [PA]' XGRD YGRD
COLO ROUG NOIR
LIST 51 61 AXES 1.0 'PRESS. [PA]'
TRAC 52 62 AXES 1.0 'DENS. [KG/M3]' XGRD YGRD
COLO ROUG NOIR
LIST 52 62 AXES 1.0 'DENS. [KG/M3]'
TRAC 55 65 AXES 1.0 'VELOC. [M/S]' YZER XGRD YGRD
COLO ROUG NOIR
LIST 55 65 AXES 1.0 'VELOC. [M/S]'
FIN

```

## shoc61z.epx

```

Post treatment (visualization from alice file)
ECHO
RESU SPLI ALIC 'shoc61.ali' GARD PSCR
OPTI PRIN
SORT VISU NSTO 24
FIN

```

## shoc62.dgibi

```

opti echo 1;
opti dime 3 elem cub8;
opti trac psc ftra 'shoc62_mesh.ps';
opti sauv form 'shoc62.msh';
*
dpla = 0.3;
rpla = 0.5 * dpla;
dext = 0.625;
rext = 0.5 * dext;
dtank = 1.6;
rtank = 0.5 * dtank;
ltan0 = -0.7;
ltan1 = 0.9;
ltanopen = 1.4;
ltan2 = 1.5;
*
ltubhp = 2;
ltublp = 16.20;
ltube = ltubhp + ltublp;
*

```

```

nrpla = 3;
tol = 0.01 * dpla / nrpla;
*
p0s = 0 0 0;
p1s = 0 rpla 0;
p2s = 0 rpla rpla;
p3s = 0 0 rpla;
*
c1s = p0s d nrpla p1s;
c2s = p1s d nrpla p2s;
c3s = p2s d nrpla p3s;
c4s = p3s d nrpla p0s;
plat = dall c1s c2s c3s c4s plan;
*
p0 = p0s plus p0s;
p1 = p1s plus p0s;
p2 = p2s plus p0s;
p3 = p3s plus p0s;
*
c1 = p0 d nrpla p1;
c2 = p1 d nrpla p2;
c3 = p2 d nrpla p3;
c4 = p3 d nrpla p0;
sflu1 = dall c1 c2 c3 c4 plan;
*
paxis = 1 0 0;
p4 = 0 rext 0;
p5 = p4 tour 45 p0 paxis;
p6 = 0 0 rext;
*
c5 = p1 d nrpla p4;
c6 = p4 c nrpla p0 p5;
c7 = p5 d nrpla p2;
c8 = p2 d nrpla p1;
sflu2 = dall c5 c6 c7 c8 plan;
*
c9 = p2 d nrpla p5;
c10 = p5 c nrpla p0 p6;
c11 = p6 d nrpla p3;
c12 = p3 d nrpla p2;
sflu3 = dall c9 c10 c11 c12 plan;
*
p7 = 0 rtank 0;
p8 = p7 tour 45 p0 paxis;
p9 = 0 0 rtank;
*
din = (rext - rpla) / nrpla;
dfi = din * dtank / dext;
c13 = p4 d p7 dini din dfi dfi;
sflu4 = c13 rota 3 45 p0 paxis;
*
sflu5 = sflu4 tour 45 p0 paxis;
*
sflu0 = sflu4 et sflu5;
sflu = sflu1 et sflu2 et sflu3 et sflu0;
elim tol sflu;
*
d1 = 0.05;
d2 = d1 * 2;
d3 = d2 * 2;
vtan0 = (sflu0 plus ((0 - ltube) 0 0)) volu dini d2 dfi d1
      tran ((0-ltan0) 0 0);
vtan1 = sflu volu dini d1 dfi d2 tran (ltan1 0 0);
vtanopen = (sflu plus (ltan1 0 0)) volu dini d2 dfi d3
      tran (ltanopen 0 0);
vtan2 = (sflu plus ((ltan1+ltanopen) 0 0)) volu dini d3 dfi d3
      tran (ltan2 0 0);
tank = vtan0 et vtan1 et vtanopen et vtan2;
elim tol tank;
*
pai = ltan1 rtank 0;
pa2 = (ltan1 + ltanopen) rtank 0;
cab = d pai pa2 dini d2 dfi d3;
abso = cab rota (2*nrpla) 90 p0 paxis;
elim tol (abso et tank);
*
*tubehp = (sflu1 plus ((0 - ltube) 0 0)) volu dini d1 dfi d1
      tran (ltubhp 0 0);
*tubelp = (sflu1 plus ((0 - ltublp) 0 0)) volu dini d1 dfi d1
      tran (ltublp 0 0);
*
pid1 = (0 - ltube) 0 0;
pid2 = pid1 plus (ltubhp 0 0);
tubehp = pid1 d pid2 dini d1 dfi d1;
lenlp3d = 1.2;
pid3 = (0 - lenlp3d) 0 0;
tubelp1 = pid2 d pid3 dini d1 dfi d1;
stub3d = sflu1 plus pid3;
tubelp3 = stub3d volu dini d1 dfi d1 tran (lenlp3d 0 0);
tubelp = stubelp1 et tubelp3;
*
raclp = manu supe (pid3 et stub3d);
*
tube = tubehp et tubelp;
*
flui = tube et tank;
elim tol flui;
trac cach tank;
trac cach flui;
*
list (nbel flui);
list (nbno flui);
list (nbel tube);
list (nbel tank);

```

```

list (nbel plat);
list (nbno plat);
*
mesh = flui et plat et abso et raclp;
tass mesh noop;
sauv form mesh;
*
fin;

```

## shoc62.epx

```

SHOC62
ECHO
!CONV win
CAST mesh
TRID ALE EROS 1.0
DIME JONC 17 ! Total n. of nodes in a TUBM junction
      ADAP NPOI 115000 CUVF 120000 NVFI 380000 ENDA
      NALE 1 NBLE 1 TERM
GEOM CUVF tubelp3 tank TUVF tubehp tubelp1 Q4GS plat
! Attention: we are obliged to assign (dummy) CL3D elements to the stub3d
! object (3D part of the 1D-3D junction), because the name of this object
! is needed below in the RACC TUBM FACE directive and it seems that the
! code expects there an object made of elements, not just of nodes
! (although only the nodes are really needed, at least according
! to the manual)
      CL3D abso stub3d
      TUBM raclp TERM
COMP DIAM DROI 0.1692568 LECT tubehp tubelp1 TERM
      RACC TUBM LECT raclp TERM
      NTUB LECT pid3 TERM DTUB 0.1692568
      FACE LECT stub3d TERM COEF 1.0
      EPAI 0.8E-3 LECT plat TERM
! Attention: the TUBM element (raclp) is NOT included
! in the "mesh" object (although it IS indeed passed in from Cast3m).
! For this reason we must add it explicitly in the GROU directive below
! if we want to have it in the extracted element groups.
      GROU 4 'endtube' LECT tube TERM COND XB GT -0.6
      'trigger' LECT tube TERM COND NEAR POIN -1.01 0 0
      'frefine' LECT flui TERM COND XB GT -1.2
      COND XB LT 1.0
      'nraclp' LECT mesh raclp TERM
      COND XB GT -1.201 COND XB LT -1.199
      COUL ROUG LECT tubehp TERM
      TURQ LECT tubelp tank DIFF trigger TERM
      VERT LECT plat TERM
      ROSE LECT trigger TERM
      JAUN LECT abso TERM
      BLEU LECT raclp TERM
GRIL LAGR LECT plat TERM
      EULE LECT flui TERM
MATE GAZP RO 59.459 GAMM 1.4 CV 719.286 PINI 50.E5 PREF 1.E5
      LECT tubehp TERM
      GAZP RO 1.189 GAMM 1.4 CV 719.286 PINI 1.E5 PREF 1.E5
      LECT tubelp1 raclp TERM
! In order to obtain a printout at least of the 3D VFCCs I am obliged
! to use a different material for tubelp3 and other 3D parts, but with the
! same characteristics as the material used for tubelp1
      GAZP RO 1.189 GAMM 1.4 CV 719.286 PINI 1.E5 PREF 1.E5
      LECT tubelp3 tank _cuvf TERM
VPJC RO 2700.0 YOUN 70.0E9 NU 0.3 ELAS 97.6E6 MXIT 200
QR1 13.2E6 CR1 1458.0 QR2 10.6E6 CR2 185.9
PDOT 5.E-4 C 1.4E-2 TQ 0.9 CP 910.0
TM 893.0 M 1.0 DC 1.0 WC 44.6E6
      LECT plat TERM
      CLVF ABSO RO 1.20
      LECT abso TERM
LINK COUP
      BLOQ 123456 LECT c2s c3s TERM
!
! cont spla not yet available in adaptivity for boundary-hanging nodes!
! use explicit blockages instead
!
! CONT SPLA NX 0 NY 0 NZ 1 LECT c1s DIFF p1s TERM
! CONT SPLA NX 0 NY 1 NZ 0 LECT c4s DIFF p3s TERM
BLOQ 345 LECT c1s DIFF p0s p1s TERM
BLOQ 246 LECT c4s DIFF p0s p3s TERM
BLOQ 23456 LECT p0s TERM
LINK DECO FLSW STRU LECT plat TERM
      FLUI LECT flui TERM
      R 0.0436 ! 0.87*H_FLUID = 0.87*0.05 = 0.0435
      HGRI 0.06 ! Slightly bigger than structural element 0.05
DGRI
FACE
BFLU 2 ! block if at least one node is in influence domain
FSCP 1 ! couple in all directions
      ADAP LMAX 4 SCAL 16
ECRI DEPL VITE TFRE 1.E-3
      POIN LECT p0s TERM NOEL
      FICH SPLI ALIC FREQ 0 TFRE 0.D0
      TIME PROG 0.D0 PAS 1.D-3 30.D-3 PAS 0.1D-3 40.D-3
      PAS 1.D-3 80.D-3
OPTI NOTE CSTA 0.5
      LOG 1
      REZO GAMO 0.5
      VFCC FCON 6 ! hllc solver
      ORDR 2 ! order in space
      OTPS 2 ! order in time
!
! RECO 1 ! Not accepted by CAL_VFCC_1D
CALC TINI 0 TEND 23.E-3
FIN

```

## shoc62b.epx

```
Post treatment (visualization from alice file)
ECHO
RESU SPLI ALIC 'shoc62.ali' GARD PSCR
OPTI PRIN
SORT VISU NSTO 1
PLAY
CAME 6 EYE -1.00291E-01 -3.09784E+00 -3.09784E+00
! Q 3.82683E-01 9.23880E-01 0.00000E+00 0.00000E+00
VIEW 0.00000E+00 7.07107E-01 7.07107E-01
RIGH 1.00000E+00 0.00000E+00 0.00000E+00
UP 0.00000E+00 -7.07107E-01 7.07107E-01
FOV 2.48819E+01
!NAVIGATION MODE: ROTATING CAMERA
!CENTER : -1.00291E-01 4.00000E-01 4.00000E-01
!RSPHERE: 1.23667E+00
!RADIUS : 4.94669E+00
!ASPECT : 1.00000E+00
!NEAR : 3.71002E+00
!FAR : 7.42004E+00
CAME 5 EYE 5.61480E-02 -5.86912E-01 -5.86912E-01
! Q 3.82683E-01 9.23880E-01 0.00000E+00 0.00000E+00
VIEW 0.00000E+00 7.07107E-01 7.07107E-01
RIGH 1.00000E+00 0.00000E+00 0.00000E+00
UP 0.00000E+00 -7.07107E-01 7.07107E-01
FOV 2.48819E+01
!NAVIGATION MODE: ROTATING CAMERA
!CENTER : 5.61480E-02 7.50000E-02 7.50000E-02
!RSPHERE: 1.20011E-01
!RADIUS : 9.36084E-01
!ASPECT : 1.00000E+00
!NEAR : 8.16074E-01
!FAR : 1.17611E+00
SCEN GEOM NAVI FREE
LIMA ON
SLER CAM1 6 NFRA 1
TRAC OFFS FICH BMP OBJE LECT flui TERM REND
FREQ 23
GO
SCEN GEOM NAVI FREE
LINE HEOU
ISO FILL FIEL ECRO 1 SUPP LECT flui TERM
LIMA ON
SLER CAM1 6 NFRA 1
TRAC OFFS FICH BMP OBJE LECT flui TERM REND
ENDPLAY
FIN
```

## shoc62p.epx

```
Post-treatment (space curves from alice file)
ECHO
RESU SPLI ALIC 'shoc62.ali' GARD PSCR
COMP NGRO 1 'xaxo' LECT flui TERM
COND LINE X1 -18.2 Y1 0 Z1 0
X2 3.8 Y2 0 Z2 0 TOL 0.0001
SORT GRAP
PERF 'shoc62p.pun'
AXTE 1.0 'Time [s]'
SCOU 51 'p_21' NSTO 21 SAXE 1.0 'curr_abscissa' LECT xaxo TERM
SUPP LECT flui TERM
ECRO COMP 1
SCOU 52 'ro_21' NSTO 21 SAXE 1.0 'curr_abscissa' LECT xaxo TERM
SUPP LECT flui TERM
ECRO COMP 2
SCOU 55 'vx_21' NSTO 21 SAXE 1.0 'curr_abscissa' LECT xaxo TERM
SUPP LECT flui TERM
VCVI COMP 1
SCOU 61 'p_24' NSTO 24 SAXE 1.0 'curr_abscissa' LECT xaxo TERM
SUPP LECT flui TERM
ECRO COMP 1
SCOU 62 'ro_24' NSTO 24 SAXE 1.0 'curr_abscissa' LECT xaxo TERM
SUPP LECT flui TERM
ECRO COMP 2
SCOU 65 'vx_24' NSTO 24 SAXE 1.0 'curr_abscissa' LECT xaxo TERM
SUPP LECT flui TERM
VCVI COMP 1
TRAC 51 61 AXES 1.0 'PRESS. [PA]' XGRD YGRD
COLO ROUG NOIR
LIST 51 61 AXES 1.0 'PRESS. [PA]'
TRAC 52 62 AXES 1.0 'DENS. [KG/M3]' XGRD YGRD
COLO ROUG NOIR
LIST 52 62 AXES 1.0 'DENS. [KG/M3]'
TRAC 55 65 AXES 1.0 'VELOC. [M/S]' YZER XGRD YGRD
COLO ROUG NOIR
LIST 55 65 AXES 1.0 'VELOC. [M/S]'
FIN
```

## shoc62z.epx

```
Post treatment (visualization from alice file)
ECHO
RESU SPLI ALIC 'shoc62.ali' GARD PSCR
OPTI PRIN
SORT VISU NSTO 24
FIN
```

## shoc65.dgibi

```
opti echo 1;
opti dime 3 elem cub8;
opti trac psc ftra 'shoc65_mesh.ps';
opti sauv form 'shoc65.msh';
*
dpla = 0.3;
rpla = 0.5 * dpla;
dext = 0.625;
rext = 0.5 * dext;
dtank = 1.6;
rtank = 0.5 * dtank;
ltan0 = -0.7;
ltan1 = 0.9;
ltanopen = 1.4;
ltan2 = 1.5;
*
ltubhp = 2;
ltublp = 16.20;
ltube = ltubhp + ltublp;
*
nrpla = 3;
tol = 0.01 * dpla / nrpla;
*
p0s = 0 0 0;
p1s = 0 rpla 0;
p2s = 0 rpla rpla;
p3s = 0 0 rpla;
*
c1s = p0s d nrpla p1s;
c2s = p1s d nrpla p2s;
c3s = p2s d nrpla p3s;
c4s = p3s d nrpla p0s;
plat = dall c1s c2s c3s c4s plan;
*
p0 = p0s plus p0s;
p1 = p1s plus p0s;
p2 = p2s plus p0s;
p3 = p3s plus p0s;
*
c1 = p0 d nrpla p1;
c2 = p1 d nrpla p2;
c3 = p2 d nrpla p3;
c4 = p3 d nrpla p0;
sflul = dall c1 c2 c3 c4 plan;
*
paxis = 1 0 0;
p4 = 0 rext 0;
p5 = p4 tour 45 p0 paxis;
p6 = 0 0 rext;
*
c5 = p1 d nrpla p4;
c6 = p4 c nrpla p0 p5;
c7 = p5 d nrpla p2;
c8 = p2 d nrpla p1;
sflu2 = dall c5 c6 c7 c8 plan;
*
c9 = p2 d nrpla p5;
c10 = p5 c nrpla p0 p6;
c11 = p6 d nrpla p3;
c12 = p3 d nrpla p2;
sflu3 = dall c9 c10 c11 c12 plan;
*
p7 = 0 rtank 0;
p8 = p7 tour 45 p0 paxis;
p9 = 0 0 rtank;
*
din = (rext - rpla) / nrpla;
dfi = din * dtank / dext;
c13 = p4 d p7 dini din dfi;
sflu4 = c13 rota 3 45 p0 paxis;
*
sflu5 = sflu4 tour 45 p0 paxis;
*
sflu0 = sflu4 et sflu5;
sflu = sflul et sflu2 et sflu3 et sflu0;
elim tol sflu;
*
d1 = 0.05;
d2 = d1 * 2;
d3 = d2 * 2;
vtan0 = (sflu0 plus (ltan0 0 0)) volu dini d2 dfi d1
tran ((0-ltan0) 0 0);
vtan1 = sflu volu dini d1 dfi d2 tran (ltan1 0 0);
vtanopen = (sflu plus (ltan1 0 0)) volu dini d2 dfi d3
tran (ltanopen 0 0);
vtan2 = (sflu plus ((ltan1+ltanopen) 0 0)) volu dini d3 dfi d3
tran (ltan2 0 0);
tank = vtan0 et vtan1 et vtanopen et vtan2;
elim tol tank;
*
pa1 = ltan1 rtank 0;
pa2 = (ltan1 + ltanopen) rtank 0;
cab = d pa1 pa2 dini d2 dfi d3;
abso = cab rota (2*nrpla) 90 p0 paxis;
elim tol (abso et tank);
*
*tubehp = (sflul plus ((0 - ltube) 0 0)) volu dini d1 dfi d1
tran (ltubhp 0 0);
*tubelp = (sflul plus ((0 - ltublp) 0 0)) volu dini d1 dfi d1
tran (ltublp 0 0);
pid1 = (0 - ltube) 0 0;
```

```

pid2 = pid1 plus (ltubhp 0 0);
tubehp = pid1 d pid2 dini d1 dfin d1;
lenlp3d = 1.2;
pid3 = (0 - lenlp3d) 0 0;
tubelp1 = pid2 d pid3 dini d1 dfin d1;
stub3d = sflui plus pid3;
tubelp3 = stub3d volu dini d1 dfin d1 tran (lenlp3d 0 0);
tubelp = tubelp1 et tubelp3;
*
raclp = manu supe (pid3 et stub3d);
*
tube = tubehp et tubelp;
*
flui = tube et tank;
elim tol flui;
trac cach tank;
trac cach flui;
*
list (nbel flui);
list (nbno flui);
list (nbel tube);
list (nbel tank);
list (nbel plat);
list (nbno plat);
*
mesh = flui et plat et abso et raclp;
tass mesh noop;
sauv form mesh;
*
fin;

```

## shoc65.epx

```

SHOC65
ECHO
!CONV win
CAST mesh
TRID ALE EROS 1.0
DIME JUNC 17 ! Total n. of nodes in a TUBM junction
ADAP NPOI 115000 CUVF 120000 NVFI 380000 ENDA
NALE 1 NBLE 1 TERM
GEOM CUVF tubelp3 tank TUVF tubehp tubelp1 Q4GS plat
! Attention: we are obliged to assign (dummy) CL3D elements to the stub3d
! object (3D part of the 1D-3D junction), because the name of this object
! is needed below in the RACC TUBM FACE directive and it seems that the
! code expects there an object made of elements, not just of nodes
! (although only the nodes are really needed, at least according
! to the manual)
CL3D abso stub3d
TUBM raclp TERM
COMP DIAM DROI 0.1692568 LECT tubehp tubelp1 TERM
RACC TUBM LECT raclp TERM
NTUB LECT pid3 TERM DTUB 0.1692568
FACE LECT stub3d TERM COEF 1.0
EPAI 0.8E-3 LECT plat TERM
! Attention: the TUBM element (raclp) is NOT included
! in the "mesh" object (although it IS indeed passed in from Cast3m).
! For this reason we must add it explicitly in the GROU directive below
! if we want to have it in the extracted element groups.
GROU 4 'endtube' LECT tube TERM COND XB GT -0.6
      'trigger' LECT tube TERM COND NEAR POIN -1.01 0 0
      'frefine' LECT flui TERM COND XB GT -1.2
                COND XB LT 1.0
      'nraclp' LECT mesh raclp TERM
                COND XB GT -1.201 COND XB LT -1.199
COUL ROUG LECT tubehp TERM
TURQ LECT tubelp tank DIFF trigger TERM
VERT LECT plat TERM
ROSE LECT trigger TERM
JAUN LECT abso TERM
BLEU LECT raclp TERM
GRIL LAGR LECT plat TERM
EULE LECT flui TERM
MATE GAZP RO 59.459 GAMM 1.4 CV 719.286 PINI 50.E5 PREF 1.E5
      LECT tubehp TERM
      GAZP RO 1.189 GAMM 1.4 CV 719.286 PINI 1.E5 PREF 1.E5
      LECT tubelp1 raclp TERM
! In order to obtain a printout at least of the 3D VFCCs I am obliged
! to use a different material for tubelp3 and other 3D parts, but with the
! same characteristics as the material used for tubelp1
GAZP RO 1.189 GAMM 1.4 CV 719.286 PINI 1.E5 PREF 1.E5
      LECT tubelp3 tank _cuvf TERM
VPJC RO 2700.0 YOUN 70.0E9 NU 0.3 ELAS 97.6E6 MXIT 200
QR1 13.2E6 CR1 1458.0 QR2 10.6E6 CR2 185.9
PDOT 5.E-4 C 1.4E-2 TQ 0.9 CP 910.0
TM 893.0 M 1.0 DC 1.0 WC 44.6E6
LECT plat TERM
CLVF ABSO RO 1.20
LECT abso TERM
LINK COUP
BLOQ 123456 LECT c2s c3s TERM
!
! cont spla not yet available in adaptivity for boundary-hanging nodes!
! use explicit blockages instead
! CONT SPLA NX 0 NY 0 NZ 1 LECT c1s DIFF p1s TERM
! CONT SPLA NX 0 NY 1 NZ 0 LECT c4s DIFF p3s TERM
BLOQ 345 LECT c1s DIFF p0s p1s TERM
BLOQ 246 LECT c4s DIFF p0s p3s TERM
BLOQ 23456 LECT p0s TERM
TBLO 126 TRIG LECT c1s DIFF p0s p1s TERM
135 TRIG LECT c4s DIFF p0s p3s TERM

```

```

1 TRIG LECT p0s TERM
123456 TRIG LECT plat DIFF c1s c2s c3s c4s TERM
LINK DECO FLSW STRU LECT plat TERM
FLUI LECT flui TERM
R 0.0436 ! 0.87*H_FLUID = 0.87*0.05 = 0.0435
HGRI 0.06 ! Slightly bigger than structural element 0.05
DGRI
FACE
BFLU 2 ! block if at least one node is in influence domain
FSCP 1 ! couple in all directions
ADAP LMAX 4 SCAL 16
ECRI DEPL VITE TFRE 1.E-3
POIN LECT p0s TERM NOEL
FICH SPLI ALIC FREQ 0 TFRE 0.D0
TIME PROG 0.D0 PAS 1.D-3 30.D-3 PAS 0.1D-3 40.D-3
PAS 1.D-3 80.D-3
OPTI NOTE CSTA 0.5
LOG 1
REZO GAMO 0.5
VFCC FCON 6 ! hllc solver
ORDR 2 ! order in space
OTPS 2 ! order in time
! RECO 1 ! Not accepted by CAL_VFCC_1D
ADAP TRIG ECRO 1 TVAL 1.02E5 LECT trigger TERM
NOCR TRIG LECT plat TERM
CALC TINI 0 TEND 23.E-3
FIN

```

## shoc65b.epx

```

Post treatment (visualization from alice file)
ECHO
RESU SPLI ALIC 'shoc65.ali' GARD PSCR
OPTI PRIN
SORT VISU NSTO 1
PLAY
CAME 6 EYE -1.00291E-01 -3.09784E+00 -3.09784E+00
! Q 3.82683E-01 9.23880E-01 0.00000E+00 0.00000E+00
VIEW 0.00000E+00 7.07107E-01 7.07107E-01
RIGH 1.00000E+00 0.00000E+00 0.00000E+00
UP 0.00000E+00 -7.07107E-01 7.07107E-01
FOV 2.48819E+01
!NAVIGATION MODE: ROTATING CAMERA
!CENTER : -1.00291E-01 4.00000E-01 4.00000E-01
!RSPHERE: 1.23667E+00
!RADIUS : 4.94669E+00
!ASPECT : 1.00000E+00
!NEAR : 3.71002E+00
!FAR : 7.42004E+00
CAME 5 EYE 5.61480E-02 -5.86912E-01 -5.86912E-01
! Q 3.82683E-01 9.23880E-01 0.00000E+00 0.00000E+00
VIEW 0.00000E+00 7.07107E-01 7.07107E-01
RIGH 1.00000E+00 0.00000E+00 0.00000E+00
UP 0.00000E+00 -7.07107E-01 7.07107E-01
FOV 2.48819E+01
!NAVIGATION MODE: ROTATING CAMERA
!CENTER : 5.61480E-02 7.50000E-02 7.50000E-02
!RSPHERE: 1.20011E-01
!RADIUS : 9.36084E-01
!ASPECT : 1.00000E+00
!NEAR : 8.16074E-01
!FAR : 1.17611E+00
SCEN GEOM NAVI FREE
LIMA ON
SLER CAM1 6 NFRA 1
TRAC OFFS FICH BMP OBJE LECT flui TERM REND
FREQ 23
GO
SCEN GEOM NAVI FREE
LINE HEOU
ISO FILL FIEL ECRO 1 SUPP LECT flui TERM
LIMA ON
SLER CAM1 6 NFRA 1
TRAC OFFS FICH BMP OBJE LECT flui TERM REND
ENDPLAY
FIN

```

## shoc65p.epx

```

Post-treatment (space curves from alice file)
ECHO
RESU SPLI ALIC 'shoc65.ali' GARD PSCR
COMP NGRO 1 'xaxo' LECT flui TERM
COND LINE X1 -18.2 Y1 0 Z1 0
X2 3.8 Y2 0 Z2 0 TOL 0.0001
SORT GRAP
PERF 'shoc65p.pun'
AXTE 1.0 'Time [s]'
SCOU 51 'p_21' NSTO 21 SAXE 1.0 'curr_abcissa' LECT xaxo TERM
SUPP LECT flui TERM
ECRO COMP 1
SCOU 52 'ro_21' NSTO 21 SAXE 1.0 'curr_abcissa' LECT xaxo TERM
SUPP LECT flui TERM
ECRO COMP 2
SCOU 55 'vx_21' NSTO 21 SAXE 1.0 'curr_abcissa' LECT xaxo TERM
SUPP LECT flui TERM
VCVI COMP 1
SCOU 61 'p_24' NSTO 24 SAXE 1.0 'curr_abcissa' LECT xaxo TERM
SUPP LECT flui TERM
ECRO COMP 1
SCOU 62 'ro_24' NSTO 24 SAXE 1.0 'curr_abcissa' LECT xaxo TERM
SUPP LECT flui TERM

```

```

                ECRO COMP 2
SCOU 65 'vx_24' NSTO 24 SAXE 1.0 'curr_abcissa' LECT xaxo TERM
                SUPP LECT flui TERM
                VCVI COMP 1
TRAC 51 61 AXES 1.0 'PRESS. [PA]' XGRD YGRD
COLO ROUG NOIR
LIST 51 61 AXES 1.0 'PRESS. [PA]'
TRAC 52 62 AXES 1.0 'DENS. [KG/M3]' XGRD YGRD
COLO ROUG NOIR
LIST 52 62 AXES 1.0 'DENS. [KG/M3]'
TRAC 55 65 AXES 1.0 'VELOC. [M/S]' YZER XGRD YGRD
COLO ROUG NOIR
LIST 55 65 AXES 1.0 'VELOC. [M/S]'

```

```

FIN

```

## shoc65z.epx

---

```

Post treatment (visualization from alice file)
ECHO
RESU SPLI ALIC 'shoc65.ali' GARD PSCR
OPTI PRIN
SORT VISU NSTO 24
FIN

```

## List of input files

### Symbols

1d3d01.dgibi.....	24	shoc61.epx.....	36
1d3d01.epx.....	24	shoc61b.epx.....	37
1d3d01p.epx.....	24	shoc61p.epx.....	37
1d3d02.dgibi.....	24	shoc61z.epx.....	37
1d3d02.epx.....	24	shoc62.dgibi.....	37
1d3d02p.epx.....	24	shoc62.epx.....	38
1d3d03.dgibi.....	25	shoc62b.epx.....	38
1d3d03.epx.....	25	shoc62p.epx.....	39
1d3d03p.epx.....	25	shoc62z.epx.....	39
1d3d14.dgibi.....	25	shoc65.dgibi.....	39
1d3d14.epx.....	25	shoc65.epx.....	40
1d3d14p.epx.....	25	shoc65b.epx.....	40
1d3d15.dgibi.....	26	shoc65p.epx.....	40
1d3d15.epx.....	26	shoc65z.epx.....	41
1d3d15p.epx.....	26		
1d3d16.dgibi.....	26		
1d3d16.epx.....	27		
1d3d16p.epx.....	27		
1d3d17.dgibi.....	27		
1d3d17.epx.....	27		
1d3d17p.epx.....	28		

### S

shoc50.dgibi.....	28
shoc50.epx.....	28
shoc50b.epx.....	29
shoc50p.epx.....	29
shoc50z.epx.....	29
shoc51.dgibi.....	29
shoc51.epx.....	30
shoc51b.epx.....	30
shoc51p.epx.....	30
shoc51z.epx.....	31
shoc52.dgibi.....	31
shoc52.epx.....	31
shoc52b.epx.....	32
shoc52p.epx.....	32
shoc52z.epx.....	32
shoc53.dgibi.....	32
shoc53.epx.....	33
shoc53b.epx.....	33
shoc53p.epx.....	33
shoc54.dgibi.....	33
shoc54.epx.....	34
shoc55.dgibi.....	34
shoc55.epx.....	35
shoc55b.epx.....	35
shoc55p.epx.....	36
shoc55z.epx.....	36
shoc61.dgibi.....	36

Europe Direct is a service to help you find answers to your questions about the European Union

Free phone number (\*): 00 800 6 7 8 9 10 11

(\*) Certain mobile telephone operators do not allow access to 00 800 numbers or these calls may be billed.

A great deal of additional information on the European Union is available on the Internet.

It can be accessed through the Europa server <http://europa.eu>

#### **How to obtain EU publications**

Our publications are available from EU Bookshop (<http://bookshop.europa.eu>), where you can place an order with the sales agent of your choice.

The Publications Office has a worldwide network of sales agents.

You can obtain their contact details by sending a fax to (352) 29 29-42758.

## JRC Mission

As the Commission's in-house science service, the Joint Research Centre's mission is to provide EU policies with independent, evidence-based scientific and technical support throughout the whole policy cycle.

Working in close cooperation with policy Directorates-General, the JRC addresses key societal challenges while stimulating innovation through developing new methods, tools and standards, and sharing its know-how with the Member States, the scientific community and international partners.

*Serving society  
Stimulating innovation  
Supporting legislation*

